Computing braids from approximate data

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

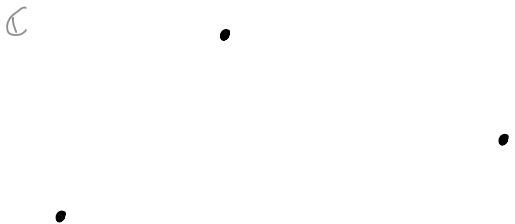
Journées Nationales de Calcul Formel
March 3, 2026 | Cirm, Marseille

Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations

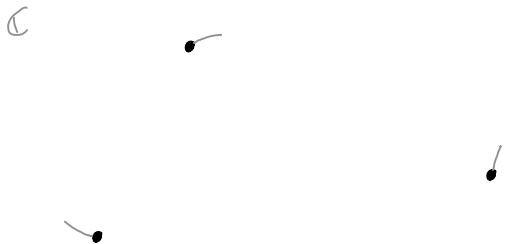


Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations

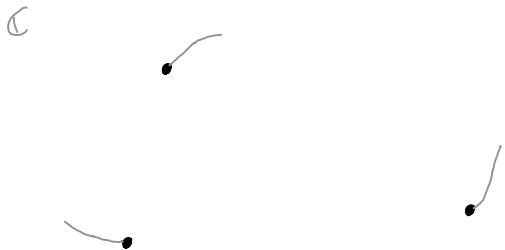


Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations

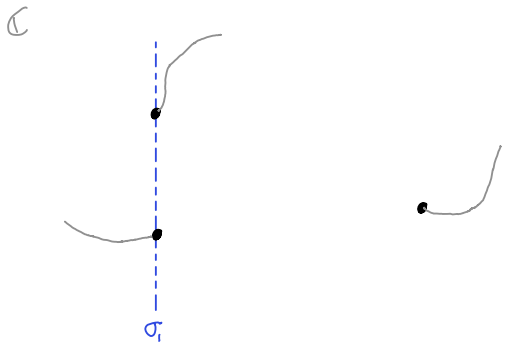


Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations

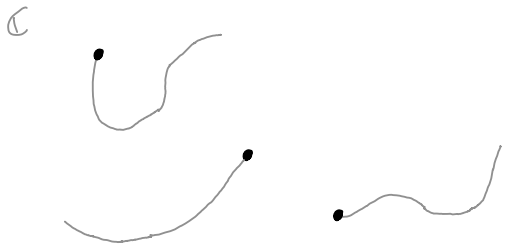


Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations



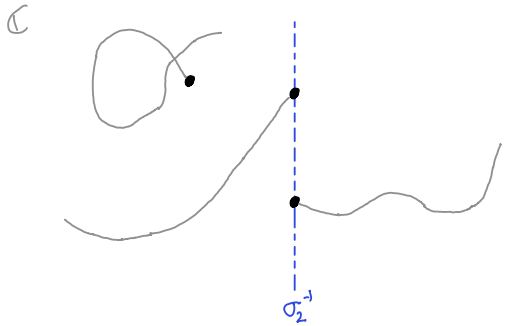
σ_1

Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations



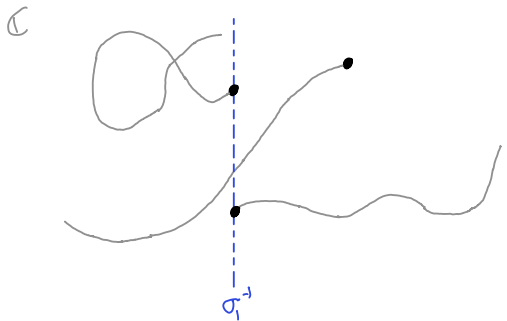
$$\sigma_1\sigma_2^{-1}$$

Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations



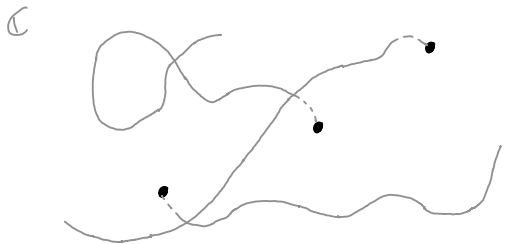
$$\sigma_1 \sigma_2^{-1} \sigma_1^{-1}$$

Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations



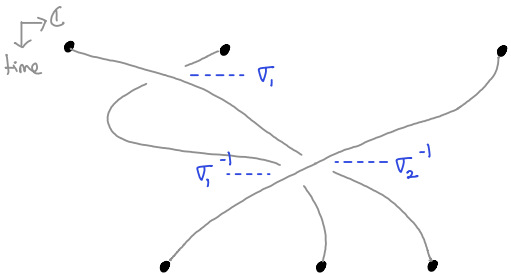
$$\sigma_1 \sigma_2^{-1} \sigma_1^{-1}$$

Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations



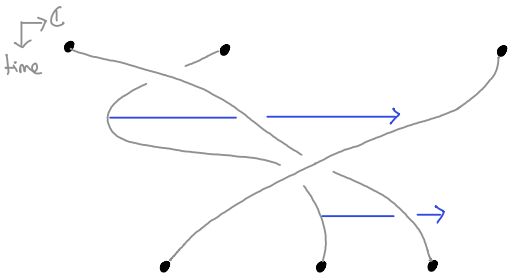
$$\sigma_1 \sigma_2^{-1} \sigma_1^{-1}$$

Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations



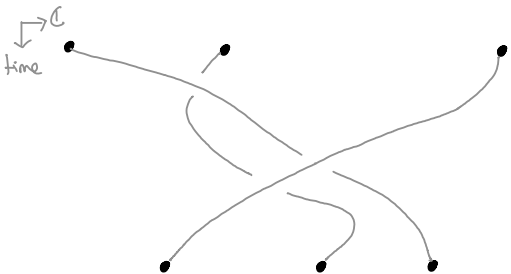
$$\sigma_1 \sigma_2^{-1} \sigma_1^{-1}$$

Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations



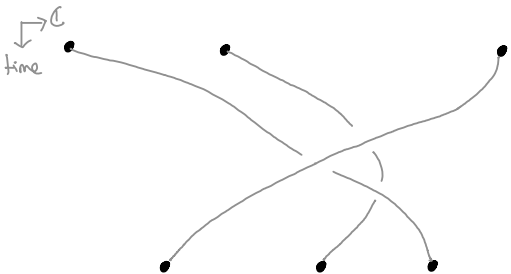
$$\sigma_1 \sigma_2^{-1} \sigma_1^{-1}$$

Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations



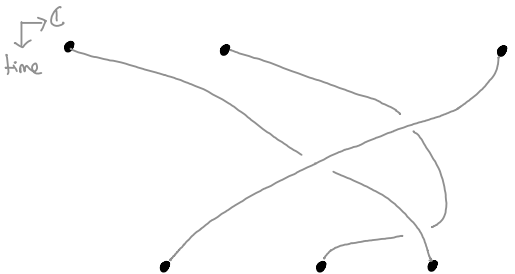
$$\sigma_1\sigma_2^{-1}\sigma_1^{-1} = \sigma_2^{-1}\sigma_1^{-1}\sigma_2$$

Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations



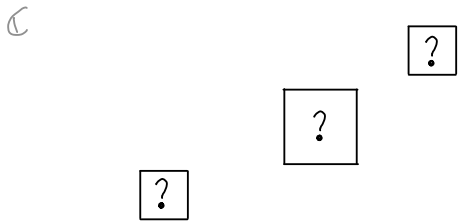
$$\sigma_1\sigma_2^{-1}\sigma_1^{-1} = \sigma_2^{-1}\sigma_1^{-1}\sigma_2$$

Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations

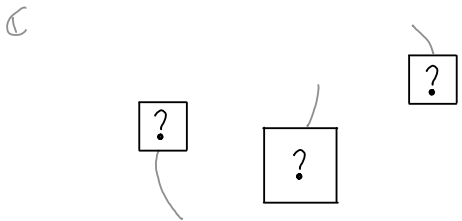


Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations

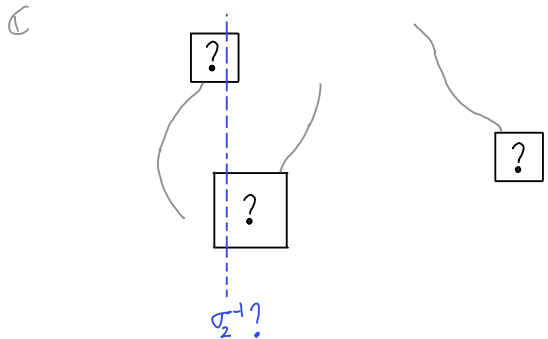


Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations



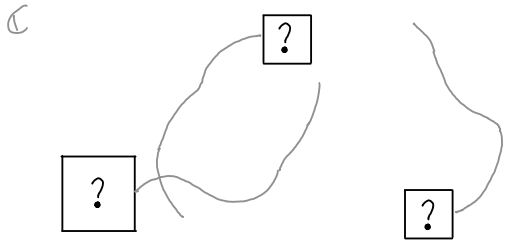
? σ_2^{-1}

Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations



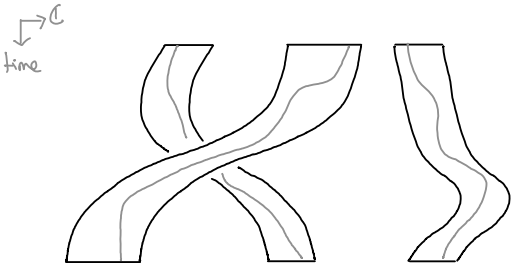
✔ σ_2^{-1}

Braids

Geometric braids $\leftrightarrow$ **Combinatorial braids**

n points moving without colliding in $\mathbb{C}$ | free group words in the letters $\sigma_1, \cdots, \sigma_{n-1}$

up to homotopy | up to explicit relations



✔ σ_2^{-1}

Motivation

Let $g \in \mathbb{C}[t][z]$. Moving t continuously in $\mathbb{C}$ induces a continuous displacement of the roots of $g(t, -)$ ❗ discriminant locus. We want to compute the associated braid.

Applications

- Fundamental group of the complement of a curve in $\mathbb{CP}^2$ [Marco-Buzunariz and Rodríguez, 2016]
- Action of the monodromy on the homology of a fibration [Pichon-Pharabod, 2024]
- Compute the topology of complex algebraic surfaces

Approach

Numerically track the roots using certified homotopy continuation.

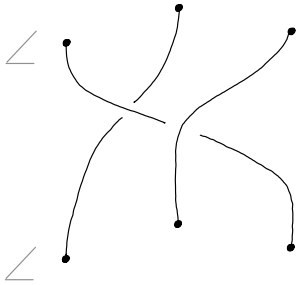
❗ We only get an isolating box for each root at all times.

Today's goal

Define $OC_n = \{(x_1, \cdots, x_n) \in \mathbb{C}^n : \text{for all } i \neq j,\ x_i \neq x_j\}$
We assume $\zeta = (\zeta_1, \ldots, \zeta_n) : [0,1] \to OC_n$. For all $t \in [0,1]$ and $i \neq j$, $\zeta_i(t) \neq \zeta_j(t)$

Goal

Input : n disjoint tubular neighborhoods around $\zeta_1, \ldots, \zeta_n$.
Output : The combinatorial braid associated to ζ.



Today's goal

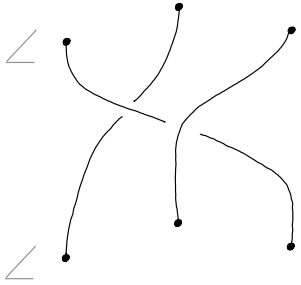
Define $OC_n = \{(x_1, \cdots, x_n) \in \mathbb{C}^n : \text{for all } i \neq j,\ x_i \neq x_j\}$
We assume $\zeta = (\zeta_1, \ldots, \zeta_n) : [0,1] \to OC_n$. For all $t \in [0,1]$ and $i \neq j$, $\zeta_i(t) \neq \zeta_j(t)$

Goal

Input : n disjoint tubular neighborhoods around $\zeta_1, \ldots, \zeta_n$.
Output : The combinatorial braid associated to ζ.

Overall strategy

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



Today's goal

Define $OC_n = \{(x_1, \cdots, x_n) \in \mathbb{C}^n : \text{for all } i \neq j,\ x_i \neq x_j\}$
We assume $\zeta = (\zeta_1, \ldots, \zeta_n) : [0,1] \to OC_n$. For all $t \in [0,1]$ and $i \neq j$, $\zeta_i(t) \neq \zeta_j(t)$

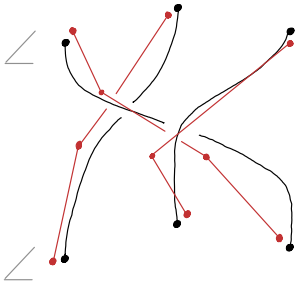
Goal

Input : n disjoint tubular neighborhoods around $\zeta_1, \ldots, \zeta_n$.
Output : The combinatorial braid associated to ζ.

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

Define $OC_n = \{(x_1, \cdots, x_n) \in \mathbb{C}^n : \text{for all } i \neq j,\ x_i \neq x_j\}$
We assume $\zeta = (\zeta_1, \ldots, \zeta_n) : [0,1] \to OC_n$. For all $t \in [0,1]$ and $i \neq j$, $\zeta_i(t) \neq \zeta_j(t)$

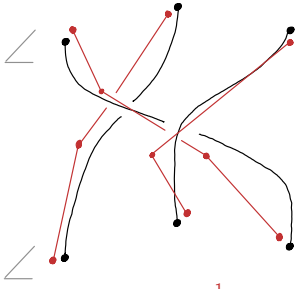
Goal

Input : n disjoint tubular neighborhoods around $\zeta_1, \ldots, \zeta_n$.
Output : The combinatorial braid associated to ζ.

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 $\tilde{\zeta}$.

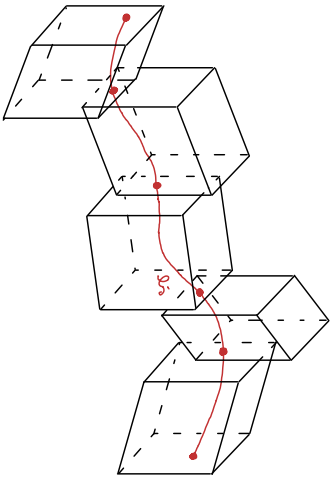


$\sigma_1 \sigma_2^{-1}$

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]

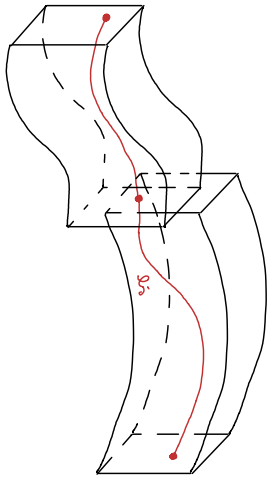
- Tubular neighborhoods are piecewise **linear**.
- For each strand ζ_i, computes a piecewise linear path in the tube.
- "Intuitive" (❗ non generic cases) algorithm on the braid with piecewise linear strands.

Algpath [Guillemot and Lairez, 2024]

- Tubular neighborhoods are piecewise **cubic**.

➕ Faster than SIROCCO.

❗ Finding a piecewise linear path in the tube requires additional work.

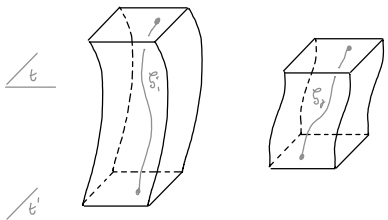


Data structure

Strand separation interface

We assume a function $\mathsf{sep}(i, j, t)$ that returns $t' \in (t, 1]$ and a symbol in $\star \in \{\rightarrow, \leftarrow, \nrightarrow, \nleftarrow\}$, 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 $\star = \nrightarrow$,
- $\mathrm{Im}(\zeta_i(s)) > \mathrm{Im}(\zeta_j(s))$ if $\star = \nleftarrow$.



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

Arrangements

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

Arrangement

Pair $(\prec_R, \prec_I)$ of partial orders on $\{1, \ldots, n\}$, such that all $i \neq j$ are comparable by $\prec_R$ or $\prec_I$.

Arrangement cell

Set of points $(x_1, \cdots, x_n) \in OC_n$ such that

- for all $i \prec_R j$, $\mathrm{Re}(x_i) < \mathrm{Re}(x_j)$,
- for all $i \prec_I j$, $\mathrm{Im}(x_i) < \mathrm{Im}(x_j)$,

In practice, we represent a partial order with a DAG.

Arrangements

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

Arrangement

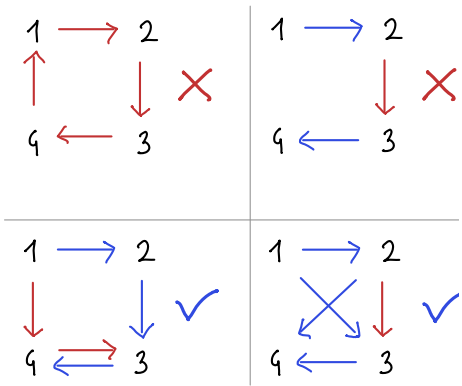
Pair $(\prec_R, \prec_I)$ of partial orders on $\{1, \ldots, n\}$, such that all $i \neq j$ are comparable by $\prec_R$ or $\prec_I$.

Arrangement cell

Set of points $(x_1, \cdots, x_n) \in OC_n$ such that

- for all $i \prec_R j$, $\mathrm{Re}(x_i) < \mathrm{Re}(x_j)$,
- for all $i \prec_I j$, $\mathrm{Im}(x_i) < \mathrm{Im}(x_j)$,

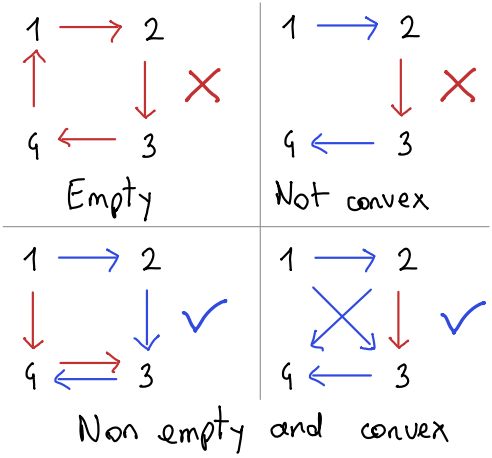
In practice, we represent a partial order with a DAG.



Properties of arrangements

Property

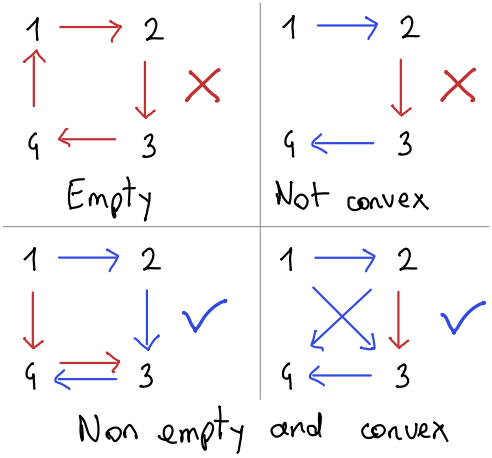
Arrangement cells are non-empty and convex subsets of OC_n.



Properties of arrangements

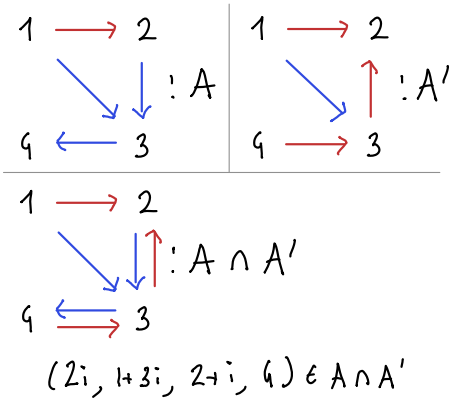
Property

Arrangement cells are non-empty and convex subsets of OC_n.



Property

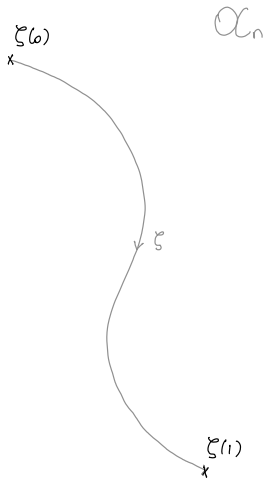
Non-empty intersection of arrangement cells is an arrangement cell.



Step 1: compute a sequence of arrangements

Path to cells

Input: ζ, given by its sep function.



Step 1: compute a sequence of arrangements

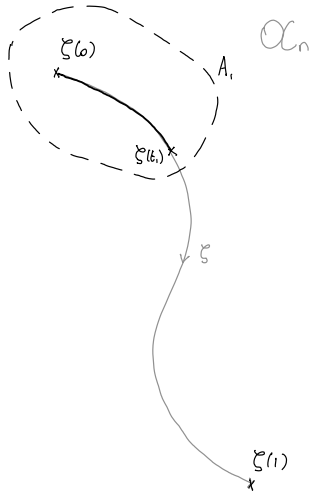
Path to cells

Input: ζ, given by its sep function.

Output: a sequence of arrangements cells $A_1, \cdots, A_r$ such that there exists $0 = t_0 < \cdots < t_r = 1$ and for any $s \in [t_{i-1}, t_i]$, $\zeta(s) \in A_i$.

Idea:

- Start with an arrangement cell A 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 arrangements

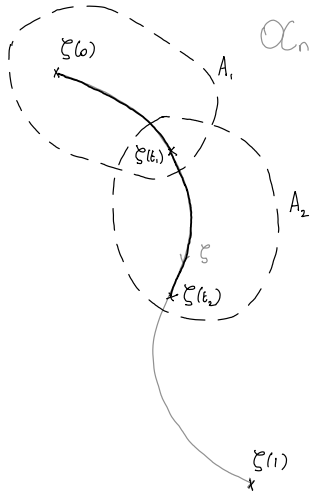
Path to cells

Input: ζ, given by its sep function.

Output: a sequence of arrangements cells $A_1, \cdots, A_r$ such that there exists $0 = t_0 < \cdots < t_r = 1$ and for any $s \in [t_{i-1}, t_i]$, $\zeta(s) \in A_i$.

Idea:

- Start with an arrangement cell A 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 arrangements

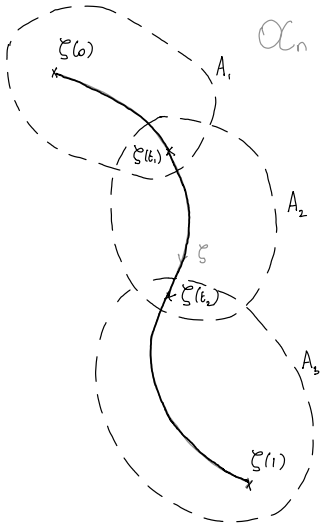
Path to cells

Input: ζ, given by its sep function.

Output: a sequence of arrangements cells $A_1, \cdots, A_r$ such that there exists $0 = t_0 < \cdots < t_r = 1$ and for any $s \in [t_{i-1}, t_i]$, $\zeta(s) \in A_i$.

Idea:

- Start with an arrangement cell A 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

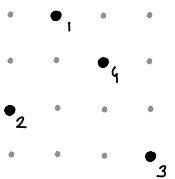
Permutation points

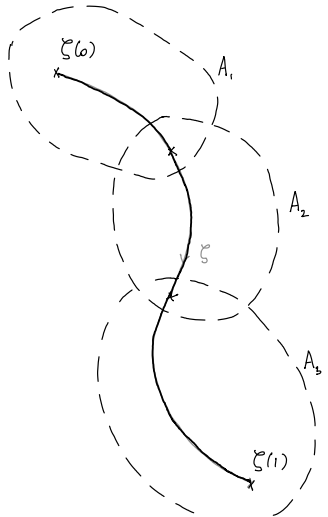
Let $\pi, \varphi \in \mathfrak{S}_n$. We define
$p_{\pi,\varphi} = (\pi(1) + \mathbf{i}\varphi(1), \cdots, \pi(n) + \mathbf{i}\varphi(n)) \in OC_n$.

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

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

$p_{\pi,e}$:





Step 2: linearize ζ

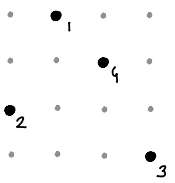
Permutation points

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

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

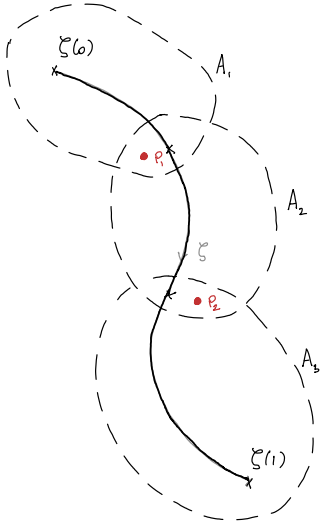
$e = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$

$p_{\pi,e}$:



Linearization of ζ

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



Step 2: linearize ζ

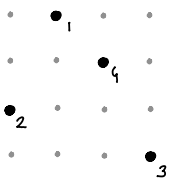
Permutation points

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

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

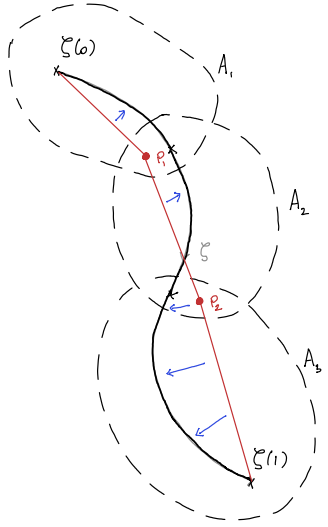
$e = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$

$p_{\pi,e}$:



Linearization of ζ

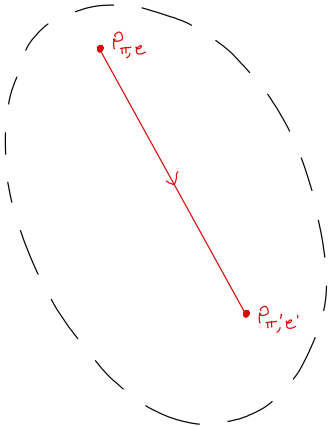
For each A_i, A_{i+1}, we compute π, φ such that $p_i = p_{\pi,\varphi}$ lies in the intersection $A_i \cap A_{i+1}$ (Hint: total order extending $\prec_R$ and $\prec_I$). The linear interpolation of the p_i is homotopic to ζ. Why? cells are convex!



Step 3: decomposition of the linearization

Reduction

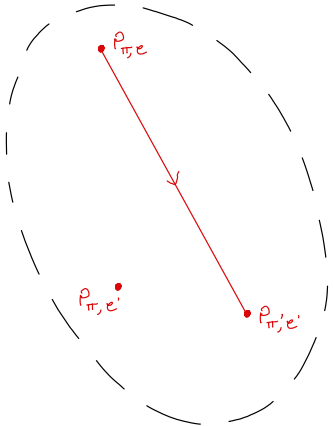
- Computing the braid associated to the whole linearization or to each piece and concatenating the results is equivalent.



Step 3: decomposition of the linearization

Reduction

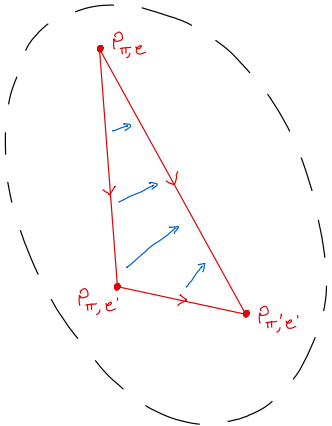
- Computing the braid associated to the whole linearization or to each piece and concatenating the results is equivalent.
- Assume $p_{\pi,\varphi}$ and $p_{\pi',\varphi'}$ both lie in an arrangement $(\prec_R, \prec_I)$. It means that π, π' extend $\prec_R$ and φ, φ' extend $\prec_I$. **So $p_{\pi,\varphi'}$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



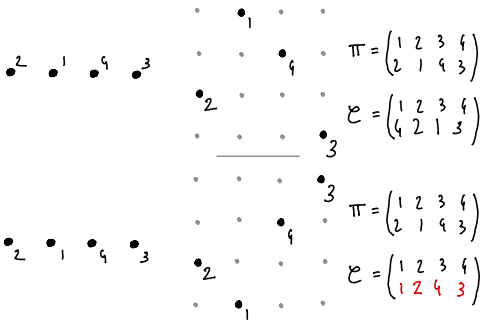
Step 3: decomposition of the linearization

Reduction

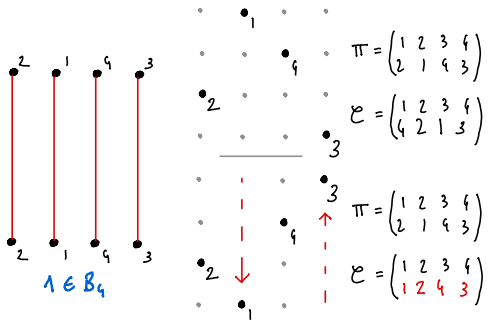
- Computing the braid associated to the whole linearization or to each piece and concatenating the results is equivalent.
- Assume $p_{\pi,\varphi}$ and $p_{\pi',\varphi'}$ both lie in an arrangement $(\prec_R, \prec_I)$. It means that π, π' extend $\prec_R$ and φ, φ' extend $\prec_I$. **So $p_{\pi,\varphi'}$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



Step 3: decomposition of the linearization

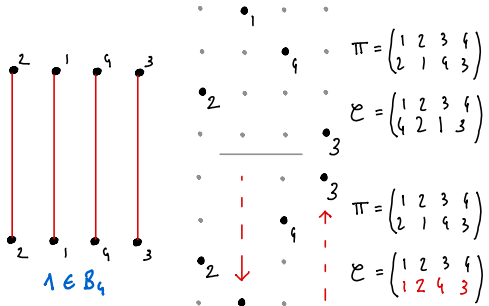


Step 3: decomposition of the linearization

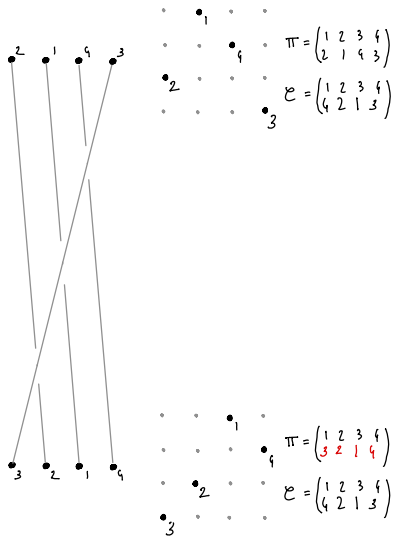


$p_{\pi,\varphi} \to p_{\pi,\varphi'}$: trivial braid.

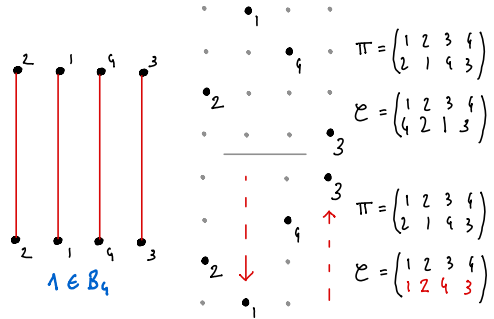
Step 3: decomposition of the linearization



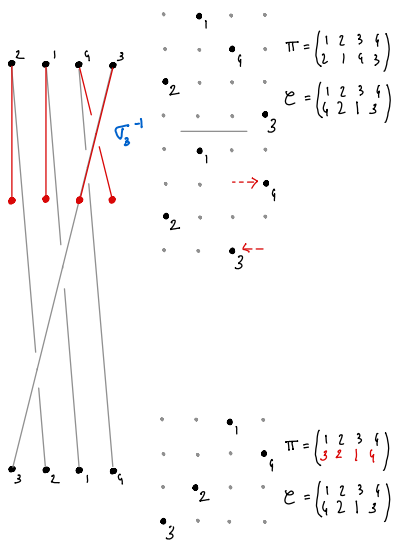
$p_{\pi,\varphi} \to p_{\pi,\varphi'}$: trivial braid.



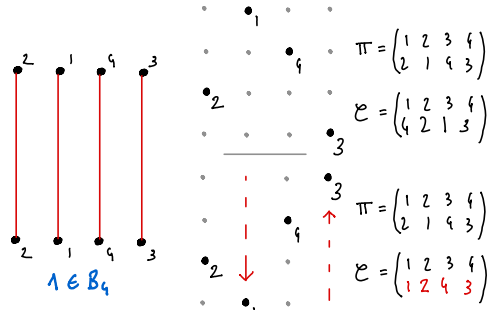
Step 3: decomposition of the linearization



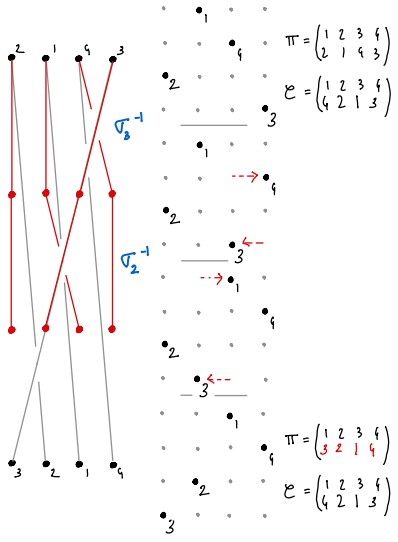
$p_{\pi,\varphi} \to p_{\pi,\varphi'}$: trivial braid.



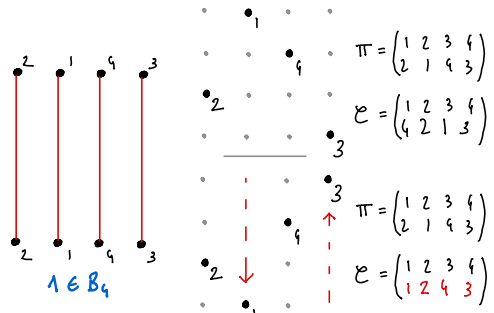
Step 3: decomposition of the linearization



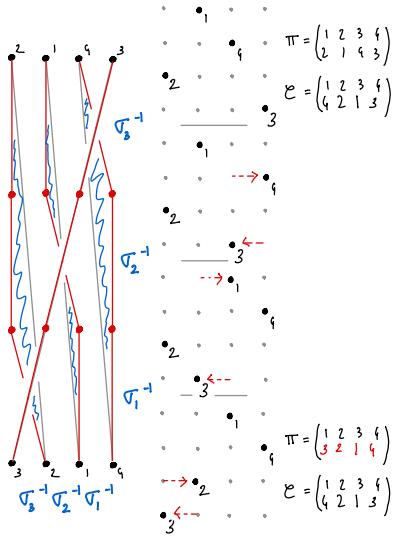
$p_{\pi,\varphi} \longrightarrow p_{\pi,\varphi'}$: trivial braid.



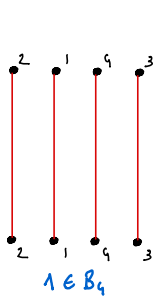
Step 3: decomposition of the linearization

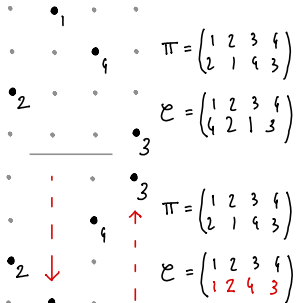


$p_{\pi,\varphi} \longrightarrow p_{\pi,\varphi'}$: trivial braid.



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'.