Computing with the braid groupoid

Intership topic

Pierre Lairez

December 9, 2022

1 Objective

Let $\widehat{\mathbb{C}} \doteq \mathbb{C} \cup \{\infty\}$ be the Riemann sphere. We consider $n+1$ points depending continuously on a parameter $t \in [0,1]$.

$$\Sigma_t \doteq \{p_0(t), \dots, p_n(t)\} \subset \mathbb{C} \tag{1}$$

that never collapse (that is $\#\Sigma_t = n+1$ for all t). Let $B_t = \widehat{\mathbb{C}} \setminus \{p_0(t), \dots, p_n(t)\}$. This moving configuration uniquely determines, for any $t \in [0,1]$, an isomorphism $\pi_1(B_0, \infty) \to \pi_1(B_t, \infty)$. In particular, if it happens that $\Sigma_0 = \Sigma_1$, the we have a automorphism of $\pi_1(B_0, \infty)$. Informally, we can realize this isomorphism as follows. We choose the standard generators for $\pi_1(B_0, \infty)$, that is paths $\gamma_1, \dots, \gamma_n$ where γ_i makes one turn around $p_i(0)$. Next, the points p_i move, the paths deform like shoelace to never touch them. For every t that gives generators of $\pi_1(B_t, \infty)$ and the map that sends γ_i to its deformation in $\pi_1(B_t, \infty)$ is the morphism. The isomorphism of $\pi_1(B_0, \infty)$ induced in this way is a special among all isomorphisms, it comes from a *braid* (Kassel & Turaev, 2008, §1.5 and §1.6). When the starting configuration is not equal to the end one, we obtain an arrow of the braid groupoid.

The main objective of the internship is to compute braids when the points p_i are described by polynomial equations. This will have important applications for the computation of the homology of complex algebraic varieties.

2 Methodology

Previous work by Rodriguez and Wang (2017) has considered this problem but their algorithm is only heuristic and I think that their methodology will not easy fit into a rigorous setting.

Let T be a rooted tree whose set of vertices is Σ. Edges are oriented toward the root. We assume that T embeds in $\mathbb{C}$: if edges are drawn with straight lines segment, they do not intersect, except at endpoints.

For an edge e of T, let $[e]_T$ denote the unique element class in $\pi_1(B, \infty)$ that is the class of a path that starts from ∞, cross e with a positive orientation, and goes back to ∞ without crossing any other edge. It is easy to check that the n edges of T freely generate $\pi_1(B)$ (see Figure 1)

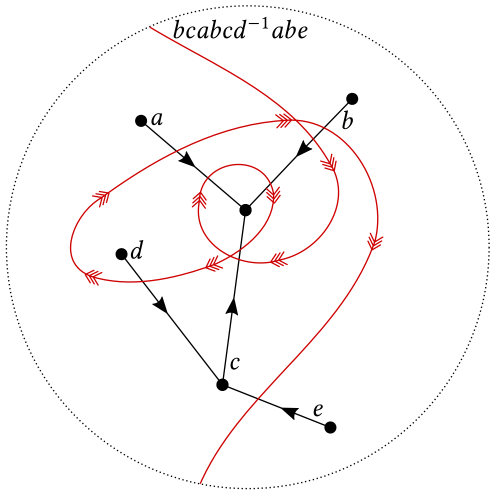

Figure 1: A path and its decomposition in terms of the generators described by a spanning tree

As the points move, the spanning tree moves along, but we want to avoid that a point crosses an edge of the tree. Therefore, from time to time, we change the tree. The generators of the fundamental group given by the new tree can be expressed in terms of the former generators, in a purely combinatorial way (see Figure 2). A spanning tree with edges labeled with words on n letters will be the main data structure in this work. Choosing and updating the tree will be made using Delaunay triangulations. Software will be developped in Julia.

References

Kassel, C., & Turaev, V. (2008). *Braid Groups* (Vol. 247). Springer. https://doi.org/10/ct8z94

Rodriguez, J. I., & Wang, B. (2017, November 21). Numerical computation of braid groups. https://doi.org/10.48550/arXiv.1711.07947

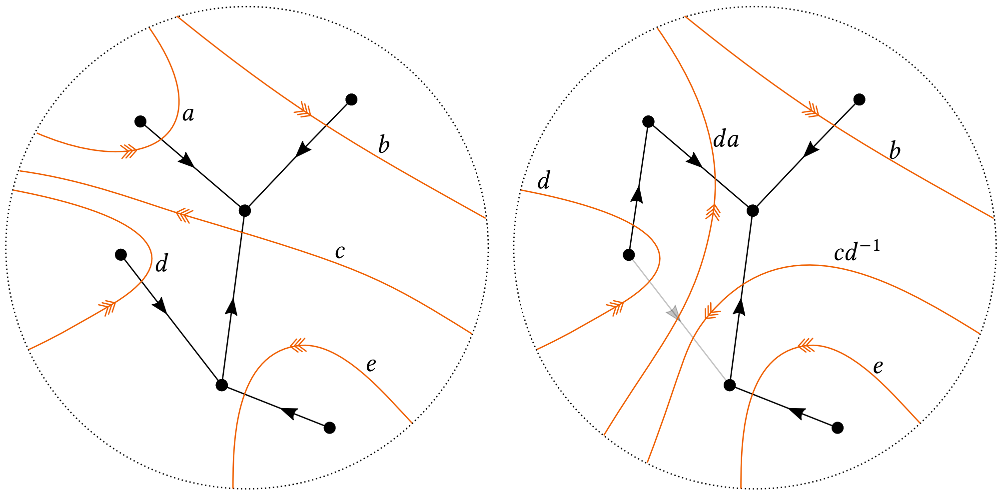

Figure 2: Paths associated to the edges of a two different spanning trees