arXiv:2601.23073v1 [cs.CG] 30 Jan 2026

从近似数据计算辫子

ALEXANDRE GUILLEMOT 和 PIERRE LAIREZ

摘要。我们研究如何根据平面中路径的近似描述来计算辫子的理论与实践问题。精确算法依赖于平面中各点的字典序排序，但这种排序在数值不确定性下并不稳定。为此，我们基于分离判定谓词，为近似数据建立了一种输入模型。例如，该模型适用于通过跟踪复系数参数化多项式的根而获得的路径，从而将经认证的路径跟踪输出与精确辫子计算联系起来。

1. 引言

当 n 个点在复平面中连续移动且彼此不碰撞时，它们就构成一个几何辫子。在同伦意义下，可以通过记录各点的绕数，以组合方式描述这类辫子，从而得到由 Artin 于 1947年 首次形式化的代数结构：辫群，其表示如下

$$B_n = \left\langle \sigma_1, \dots, \sigma_{n-1} \;\middle|\; \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}, |i-j| > 1 \Rightarrow \sigma_i\sigma_j = \sigma_j\sigma_i \right\rangle .$$

我们主要关注由参数化多项式 $f \in \mathbb{C}[t][x]$ 所产生的辫子。沿连续路径 $\gamma : [0,1] \to \mathbb{C}$ 对 t 的位移，会使 $f(t,-)$ 的根在 $\mathbb{C}$ 中连续移动。只要 γ 不接近某个临界集合——判别式簇，根就不会碰撞，因此会形成一个几何辫子。给定 f 和 γ，我们的目标是计算相应的组合辫子。这在复代数簇拓扑研究中有应用：辫子单值化可用于计算 $\mathbb{CP}^2$ 中曲线补集的基本群（Zariski，1929；Van Kampen，1933；Marco-Buzunariz 和 Rodr´ıguez，2016年），也可用于计算单值化在纤维化同调上的作用（Pichon-Pharabod，2024年）。

辫子的拓扑性质引出了一种符号数值方法：首先数值计算根的运动，然后恢复组合辫子。第一步可由经认证的同伦延拓 算法处理（van der Hoeven，2011年；Beltr´an 和 Leykin，2012、2013年；Marco-Buzunariz 和 Rodr´ıguez，2016年；Kranich，2016年；Xu 等人，2018年；Duff 和 Lee，2024年；Guillemot 和 Lairez，2024年）。这些算法会在 $\mathbb{C} \times [0,1]$ 中围绕根路径输出一组互不相交的管状邻域，这些信息足以刻画关联辫子。下文将重点讨论第二步：给定一组表示几何辫子的管状邻域，如何计算关联的组合辫子？为避免将方法局限于管状邻域的某种特定表示（不同路径跟踪算法所采用的表示在具体形式上各不相同，例如分段线性、泰勒模型等），本文采用一种抽象概念——用来表示几何辫子的近似数据。它提炼出捕捉辫子所必需的内容，形式上是一种几何谓词，可针对管状邻域的特定表示进行实现。

前人工作。 Marco-Buzunariz 和 Rodr´ıguez（2016年）解决了在给定 f and γ 的情况下计算辫子的问题，并提供了一个实现。给定通过其经认证的同伦延拓软件 SIROCCO 获得的一组管状邻域，他们会在每个管状邻域中计算分段线性路径，并据此恢复组合辫子。

日期：2026年2月2日。

本方法省去了这一步线性化。Rodriguez 和 Wang（2017年）提出了另一种方法：求解一个特定的多项式系统，其解集包含根在投影到实轴时发生交叉的时间点，然后通过同伦延拓确定交叉的符号。不过，要使该方法获得认证，并处理具有非一般交叉的几何辫，仍需进一步研究。

贡献。 我们引入一种数据结构，用于表示具有 n 股的几何辫子。该结构由一个几何谓词 *sep* 构成，它提供点的相对位置的信息，但不保留点的精确位置。更具体地说，假设点运动由 n 条路径 $F_1, \dots, F_n : [0,1] \to \mathbb{C}$ 给出，每条路径都对应一个点的运动。给定 $i, j \in [n]$ 个互不相同的点和 $t \in [0,1)$， $sep(i,j,t)$ 返回一个时间 $t' \in (t,1]$ 和一条轴 q ，该轴可以是实轴或虚轴，使得在时间区间 $[t,t']$ 内，将 F_i 投影到 q 上时，它们不会与 F_j 相交。当点运动是近似计算得到时，这种数据结构也很适用：即使无法获得点的精确位置，也可以利用误差界来确定两个点是否沿实方向或虚方向始终保持分离。对于由其 *sep* 几何谓词描述的辫子，我们提出一种算法来计算其对应的组合辫子。该方法无需在管状邻域内计算股的分段线性近似。当路径跟踪后端输出分段线性管状邻域时，线性化步骤会自动完成（Marco-Buzunariz 和 Rodr´ıguez，2016年；Duff 和 Lee，2024年）；相较之下，当管状邻域由泰勒模型给出时，线性化还需要额外处理，而泰勒模型已出现在近期的路径跟踪方法中（van der Hoeven，2011年；Guillemot 和 Lairez，2024年）。在后一种情况下，每个管状邻域由高阶片段构成，线性化可能需要多个线段，而实部或虚部的分离情况则可以在整个片段上进行检验。

2. 辫子

第 2.1 节回顾几何辫子和组合辫子的定义。第 2.2 节解释如何在精确情形下根据几何辫子计算组合辫子。

2.1. 构形空间与辫子。

对于 $n \geq 1$， 令 OC_n 表示 n 个 $\mathbb{C}$ 中点的有序构形空间：

$$OC_n = \left\{ (z_1, \dots, z_n) \in \mathbb{C}^n \;\middle|\; \forall i, j \in [n], i \neq j \Rightarrow z_i \neq z_j \right\}.$$

置换群 S_n 通过置换 z_i作用于 OC_n： 对于置换 $\sigma \in S_n$，

$$\sigma \cdot (z_1, \dots, z_n) = (z_{\sigma^{-1}(1)}, \dots, z_{\sigma^{-1}(n)}).$$

C 中 n 个点的（无序）构形空间是商空间 $C_n = OC_n/S_n$，并赋予商拓扑。

几何辫子是路径 $F : [0,1] \to C_n$的同伦类。固定 $[n] = \{1, \dots, n\} \in C_n$作为基点。满足 $F(0) = F(1) = [n]$的几何辫子 F定义了基本群 $\pi_1(C_n, [n])$ 的一个元素。Artin（1947年）引入了辫群

$$B_n = \left\langle \sigma_1, \dots, \sigma_{n-1} \;\middle|\; \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}, |i-j| > 1 \Rightarrow \sigma_i\sigma_j = \sigma_j\sigma_i \right\rangle$$

并证明了 $B_n \simeq \pi_1(C_n, [n])$。我们将 B_n的元素称为组合辫，以区别于几何辫。该同构通过将生成元 σ_i 映射到交换 i 与 $i+1$的路径给出，其中右点从左点上方经过（图1）。

即使几何辫子并非从基点 $[n]$ 出发或以其结束，仍然可以自然地为其关联一个组合辫子。为此，我们如下定义从 $[n]$ 到 c 的典范路径 R_c，适用于 C_n 中的任意构型 $c = \{z_1, \dots, z_n\}$：将点 z_i

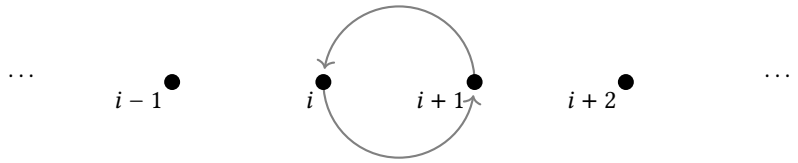


图1. 路径 C_n 诱导组合辫子 σ_i。

按字典序排列，先比较实部，若相同则比较虚部。路径 R_c 由线性插值得到

$$R_c(t) = \{(1-t)i + tz_i \mid i \in [n]\}\,.$$

对于每个 $t \in [0,1]$，这些点都按字典序严格排列，因此不会发生碰撞。给定一个几何辫子 $F : [0,1] \to C_n$，我们考虑路径拼接[1] $\widetilde{F} = R_{F(0)} \cdot F \cdot R_F^{-1}(1)$，它定义了 $\pi_1(C_n, [n])$中的一个元素，因此通过Artin同构诱导出一个组合辫子。这种约定与路径拼接相容：对于 C_n 中的任意两条路径 $\widetilde{F}$ 和 G，若 $F(1) = G(0)$，则路径 $F \cdot G$与 $F \cdot G$同伦。

2.2. 在精确情形下计算辫子。 为了更好地理解近似情形，先考虑这样一个问题：如何计算与几何辫子的精确表示相对应的组合辫子。连续函数 $[0,1] \to C_n$有许多可能的表示方式；对于计算辫子，一种实用的方式是采用分段线性表示：选取点 $p_0, \ldots, p_r \in \mathbb{C}^n$ （或者更确切地说，选取 $\mathbb{Q}[\mathrm{i}]^n$，以便执行精确算术），并通过线性插值定义函数 $F : [0,1] \to \mathbb{C}^n$：

$$F(t) = (i + 1 - rt)\, p_i + (rt - i)\, p_{i+1}, \quad \tfrac{i}{r} \le t \le \tfrac{i+1}{r}.$$

假设该函数的 n 坐标始终互不相同；那么 $F : [0,1] \to OC_n$ 是一条路径。该数据结构是精确的，可以轻松计算：

- 对任意 $t \in [0,1]$，求出唯一的置换 $\mathrm{ord}_F(t)$，使得 $\mathrm{ord}_F(t) \cdot F(t)$ 按字典序排列；
- ord_F[2] 的不连续点；
- 不连续点右侧 $\mathrm{ord}_F(t)$ 的值。

由此得到算法 1。在一般情形下，即不存在多重交叉（任意时刻至多有两个点的实部相同），证明很直接，因为总能在第 8 行的分解中选择 $s = 1$。在非一般情形下，证明则更为复杂；由于这不是本文的重点，我们省略证明。

当几何辫子 F 仅以近似方式指定时（例如通过管状邻域指定），排序函数 ord_F 的定义并不充分，因此无法再计算。一个解决方案是计算一条精确的分段线性路径 $G : [0,1] \to OC_n$，使得：

(1) $F(0)$ 中坐标的字典序（分别对应 $F(1)$）与 $G(0)$（分别对应 $G(1)$）中的坐标字典序相同；(2) 路径 $F(0) \to G(0) \xrightarrow{G} G(1) \to F(1)$（首尾两条箭头为直线）与 F 同伦。

在这种情况下，由 F和 G诱导的辫子是相同的，并且算法 1 适用于 G。这正是 Marco-Buzunariz 与 Rodríguez（2016年）所采用的方法。但这仍然需要对 $F(0)$和 $F(1)$进行字典序排序，而这在数值环境中可能无法获得。

[1]按照最常见的约定（例如 Hatcher，2002），路径 f 与路径 g 的路径拼接 $f \cdot g$ 会先沿 f 行进，然后沿 g 行进。[2]只需计算不连续点的一个有限超集，例如对所有 $1 \le i < j \le n$ 取满足 $\mathrm{Re}(F_i(t)) = \mathrm{Re}(F_j(t))$ 的所有 $t \in [0,1]$ 所成集合的边界之并。

算法 1 精确情形下的辫子计算

定义 辫子(F)**:**
  $T \leftarrow \mathrm{ord}_F$ 的所有不连续点（或这些点的超集）
  $b \leftarrow 1$ # 平凡辫子
  $\alpha \leftarrow \mathrm{ord}_F(0)$
  按 $t \in T$ 递增顺序**:**
    # 循环不变式：$\alpha = \lim_{s \to t-} \mathrm{ord}_F(s)$
    $\beta \leftarrow \lim_{s \to t+} \mathrm{ord}_F(s)$ 不连续点右侧的值
    将 $\beta\alpha^{-1}$ 写成对换的乘积（$j_s\ j_s+1$）$\cdots$（$j_1\ j_1+1$）
    对于 i，**从** 1 **到** s**:**
      $\varepsilon \leftarrow$符号 $\left(\mathrm{Im}\left(F_{\alpha^{-1}(j_i+1)}(t) - F_{\alpha^{-1}(j_i)}(t)\right)\right)$
      $b \leftarrow b \cdot \sigma_{j_i}^{\varepsilon}$ # 更新辫子
      $\alpha \leftarrow (j_i\ j_i+1) \cdot \alpha$ # 更新辫股次序
  返回 b

从实际角度来看，即使端点的排序并不含糊，计算这样一条路径 G仍可能很困难，例如当 F 以泰勒模型的管状邻域给出时；这种情况出现在经认证的路径跟踪的背景下（Guillemot & Lairez，2024年；van der Hoeven，2011年）。在管状邻域内逼近一条非线性路径所需的线段数目，会随着管半径的减小而变得无界。

3. 近似数据和排列

第 3.1 节介绍了<style id='1'>路径近似的定义，为从近似数据计算辫子提供基本背景。第 3.2 节研究了<style id='3'>排列，这是一种组合抽象，用于表示平面中 n 点的位置。排列序列被用作辫子的中间表示，介于几何辫子和组合辫子之间。

3.1. 应对数值歧义。

由路径 $F : [0,1] \to \mathbb{C}^n$诱导的辫子是一个拓扑对象，在形变下保持不变量，因此应当能够从近似数据中计算出来。但拥有近似数据究竟意味着什么？具体的数值并不重要；目标只是将各点分离并捕捉它们的绕数。为此，只需知道，对于任意 $t \in [0,1]$ 以及任意一对 $1 \le i < j \le n$，存在一个坐标轴 Re 或 Im，使得 $F_i(t)$ 和 $F_j(t)$ 沿该轴分离，同时知道它们沿此轴的相对次序。这就引出了以下定义。

定义1（路径近似）。 路径的近似 $F : [0,1] \to OC_n$ 是由两族开子集（R_{ij}）和（I_{ij}）组成的数据，它们属于 $[0,1]$,，并以互不相同的 $i,\ j \in [n]$为索引，使得对于所有此类索引对，

（i）对于所有 $t \in R_{ij}$，$\mathrm{Re}(F_i(t)) < \mathrm{Re}(F_j(t))$；（ii）对于所有 $t \in I_{ij}$，$\mathrm{Im}(F_i(t)) < \mathrm{Im}(F_j(t))$；（iii） $R_{ij} \cup R_{ji} \cup I_{ij} \cup I_{ji} = [0,1]$。

在计算层面，分离查询是获取路径近似的唯一途径：给定 t和一对 $1 \le i < j \le n$，我们希望至少找到 $\{R_{ij}, R_{ji}, I_{ij}, I_{ji}\}$中的一个元素，使其包含 t；根据（iii），这样的元素必然存在。由此得到定义1的以下面向计算的变体；本文其余部分均采用这一变体。这两个定义是等价的，因为一个定义下的近似可以给出另一个定义下的近似。

定义2（路径近似，计算变体）。一个路径的近似 F $[0,1] \to OC_n$ 是指，对于每个不同的 $i,\ j \in [n]$，给出一个可计算映射

$$sep_F(i,j,-) : [0,1] \longrightarrow [n]^2 \times \{\mathrm{Re},\mathrm{Im}\} \times [0,1]$$

使得对于所有 $t \in [0,1]$，值 $(i',\ j',\ q,\ t') = sep_F(i,\ j,\ t)$ 满足：
(sep_a) $(i',\ j') = (i,\ j)$ 或 $(j,\ i)$；(sep_b) 对所有 $s \in [t,t']$, $q(F_{i'}(s))$ $< q(F_{j'}(s))$ 均成立；(sep_c) $t' \geq \min(1,\ t+\varepsilon_F)$，其中某个 $\varepsilon_F > 0$ 仅取决于 F。

3.2. **排列**。排列是一种组合结构，用于表示复平面中各点的相对位置，并反映路径近似所提供的信息（定义1和2）。

定义3（排列，排列胞腔）。一个关于 n 点的排列 是由两个严格偏序[3]（定义在 $[n]$ 上）组成的有序对 $(\prec_{\mathrm{Re}},\ \prec_{\mathrm{Im}})$，且任意两个不同的 $i,\ j \in [n]$ 在其中至少一个偏序下可比较。与排列 $(\prec_{\mathrm{Re}},\ \prec_{\mathrm{Im}})$ 关联的排列胞腔 是由下式定义的子集 $A \subset OC_n$

$$A = \Big\{(z_1,\ldots,z_n) \in \mathbb{C}^n \;\Big|\; i \prec_{\mathrm{Re}} j \Rightarrow \mathrm{Re}(z_i) < \mathrm{Re}(z_j) \text{ and } i \prec_{\mathrm{Im}} j \Rightarrow \mathrm{Im}(z_i) < \mathrm{Im}(z_j) \text{ for all } i,j \in [n]\Big\}.$$

引理 4. 排列胞腔是 OC_n 的非空凸开子集，并且它们覆盖 OC_n。此外，排列胞腔的任何非空交集本身也是一个排列胞腔。

证明. 设 $(\prec_{\mathrm{Re}},\prec_{\mathrm{Im}})$ 为一个排列，并记 A 为其胞腔。定义A 的不等式是严格的，因此 A 在 $\mathbb{C}^n$ 中是开的，并且它们是实线性的，因此 A 是凸的。

为了看出 A 非空，选取偏序$\prec_{\mathrm{Re}}$ 和 $\prec_{\mathrm{Im}}$的线性扩张 $\prec_{\mathrm{Re}}$ 和 $\overline{\prec}_{\mathrm{Im}}$，并选取 $[n]$ 的置换 π 和 ϕ，使得 $\pi^{-1}(1) \underline{\prec}_{\mathrm{Re}} \cdots \underline{\prec}_{\mathrm{Re}} \pi^{-1}(n)$ 且 $\phi^{-1}(1) \prec_{\mathrm{Im}} \cdots \prec_{\mathrm{Im}} \phi^{-1}(n)$。构型 $\big(\pi(k) + \phi(k)\mathbf{i}\big)_{1\leq k\leq n}$ 属于 A。

包含关系 $A \subseteq OC_n$ 是直接的：给定 $z \in A$ 和 $i \neq j$，排列的定义要求 $i \prec_{\mathrm{Re}} j$、$j \prec_{\mathrm{Re}} i$、$i \prec_{\mathrm{Im}} j$或 $j \prec_{\mathrm{Im}} i$之一成立，这意味着要么 $\mathrm{Re}(z_i) \neq \mathrm{Re}(z_j)$，要么 $\mathrm{Im}(z_i) \neq \mathrm{Im}(z_j)$，因此 $z_i \neq z_j$。

对于覆盖性质，设 $z \in OC_n$，并在 $\mathrm{Re}(z_i) < \mathrm{Re}(z_j)$ 时定义 $i \prec_{\mathrm{Re}} j$，在 $\mathrm{Im}(z_i) < \mathrm{Im}(z_j)$ 时定义 $i \prec_{\mathrm{Im}} j$。z的不同点在实部或虚部上不同，因此每一对在至少一个序中变得可比较。因此（$\prec_{\mathrm{Re}},\prec_{\mathrm{Im}}$）是一个排列，其胞腔按构造包含 z。

最后，考虑两个排列（$\prec_{\mathrm{Re}},\prec_{\mathrm{Im}}$）和（$\prec'_{\mathrm{Re}},\prec'_{\mathrm{Im}}$），其单元为 A 和 A'。若存在某些 i 和 j 满足 $i \prec_{\mathrm{Re}} j$ 和 $j \prec'_{\mathrm{Re}} i$，则 $A \cap A' = \varnothing$，对 $\prec_{\mathrm{Im}}$ 和 $\prec'_{\mathrm{Im}}$亦然。因此，若 $A \cap A' \neq \varnothing$，则二元关系 $\prec_{\mathrm{Re}} \cup \prec'_{\mathrm{Re}}$ 和 $\prec_{\mathrm{Im}} \cup \prec'_{\mathrm{Im}}$ 是无环的，它们的传递闭包给出偏序。交集 $A \cap A'$ 是附属于这一对序的排列胞腔。□

定义 5（覆盖）。排列胞腔的有限序列 $(A_1,\ldots,A_r)$覆盖路径$F : [0,1] \to OC_n$，当且仅当存在一个划分 $0 = t_0 \leq t_1 \leq \cdots \leq t_r = 1$，使得 $F([t_{i-1},t_i]) \subseteq A_i$ 对每个 i都成立。

由于 OC_n 是由排列胞腔覆盖的 $\mathbb{C}^n$ 的开子集，$[0,1]$ 的紧致性保证 OC_n 中的每条路径都有一个有限覆盖序列。

回顾一下，若存在连续映射 $H : [0,1]^2 \to OC_n$，满足 $H(0,t) = F(t)$、$H(1,t) = G(t)$，且对所有 s,t，均有 $H(s,0) = F(0) = G(0)$、$H(s,1) = F(1) = G(1)$，则称两条路径 $F,G : [0,1] \to OC_n$ 同伦。下一个引理指出，路径的同伦类由其端点以及覆盖该路径的任意排列序列共同决定。

引理 6. 设 F、$G : [0,1] \to OC_n$ 是两条路径。假设

[3]一种严格偏序 是传递的、反自反的二元关系。

算法 2 计算覆盖一条路径的排列序列

输入： 一条路径 $F : [0,1] \to OC_n$ 的近似，采用 *sep* 方法给出。
输出： 覆盖 F 的排列序列。

```
def 覆盖 (F):
    A ← {} # 表示当前排列的图的边集
    生存期 ← {} # 将 A 的边映射到其生存期
    ℒ ← [] # 覆盖 F的排列列表
    t ← 0
    当 t < 1时:
        # 补全该图，直到它表示一个包含 F(t) 的排列。
        当 A 未定义排列时:
            i, j ←一个不可比较对
            e, t′ ← sep(F, i, j, t)# e ∈ [n]² × {实部，虚部}
            A ← A ∪ {e} # i 和 j 现在可以比较
            生存期[e] ← t′
        将 A 的副本追加到 L
        # 推进时间，直到某条边到期。
        e ← 生存期最短的 A 的边
        A ← A ∖ {e}
        t ←生存期[e]
    返回 ℒ
```

(i) $F(0) = G(0)$ 以及 $F(1) = G(1)$；并且 *(ii)* 存在一个排列胞腔序列，覆盖 F 和 G。则 F 和 G 同伦。

证明。 设 $(A_1, \dots, A_r)$ 是一个公共覆盖序列，并选择划分 $0 = t_0 \leq \cdots \leq t_r = 1$，使其对 F 实现覆盖；选择划分 $0 = s_0 \leq \cdots \leq s_r = 1$，使其对 G 实现覆盖。在同伦意义下，可以对 F 和 G 重新参数化，使得 $s_i = t_i$ 和不等式均为严格不等式。

定义 $H : [0,1]^2 \to \mathbb{C}^n$ 为 $H(s,t) = (1-s)F(t) + sG(t)$。这是 F与 G之间的一个同伦：显然满足边界条件，因此只需检查 $H(s,t)$ 始终位于 OC_n内。固定 $t \in [0,1]$并选择满足 $t_{i-1} \leq t \leq t_i$的 i。根据构造， $F(t)$ 和 $G(t)$ 均属于 A_i，且 A_i是凸的；因此，对所有 s， $H(s,t) \in A_i \subset OC_n$。□

算法 2 展示了如何根据路径近似，实现排列覆盖序列的计算。图 2 给出了一个示例。在此例中，该算法生成一个由 5 个排列构成的序列。排列表示为 $[n]^2 \times$ {Re,Im} 的子集，我们将其解释为顶点集为 $[n]$ 、边由 Re 或 Im 标记的图。偏序 $<_{\mathrm{Re}}$ 和 $<_{\mathrm{Im}}$ 是可达关系，分别限制在 Re 标记边或 Im 标记边上。映射 生存期 存储每条边有效的时间。

命题7. 算法 *2* 终止，并输出一个覆盖 F 的序列。

证明。 为证明终止性，设 $\varepsilon > 0$ 由条目（*sep*c）给出，并对每个整数 $k \geq 0$，令n_k 表示 A 中生存期落在 $[k\varepsilon, (k+1)\varepsilon)$ 内的边数。考察每次迭代结束时的元组（n_k）$_{0 \leq k \leq \frac{1}{\epsilon}}$，可得一个关于字典序严格递减的序列。因此，迭代次数有限。

正确性依赖于由 *sep* 的定义性质导出的如下不变量：每当 (i,j,q) 属于生存期为 t' 的 A 时，不等式 $q(F_i(s)) < q(F_j(s))$ 对所有 $s \in [t, t']$ 成立。□

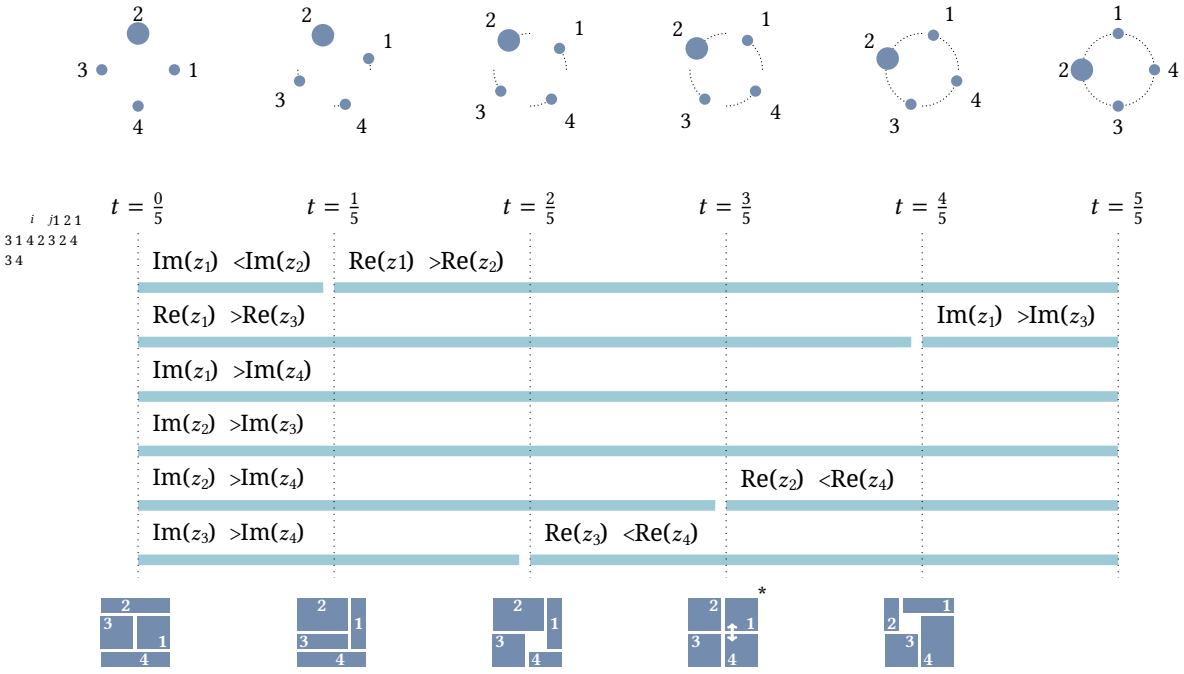


图 2. 算法 2 的示意图：四个点沿圆周运动，其位置仅近似已知（以蓝色圆盘表示）。条形表示当前排列所对应的图中一条边的生存期。最后一行的图表示所得的覆盖。标有 $*$ 的排列胞腔不能表示为盒子的构型，它由

$$\{\max(\mathrm{Re}(z_2),\mathrm{Re}(z_3)) < \min(\mathrm{Re}(z_1),\mathrm{Re}(z_4)), \mathrm{Im}(z_3) < \mathrm{Im}(z_2) \text{ and } \mathrm{Im}(z_4) < \mathrm{Im}(z_1)\}.$$

精确情形下点的组合结构比近似情形更简单。Fox 和 Neuwirth（1962）描述了 OC_n上的 CW复形结构，其胞腔为构型提供了组合描述。例如，对于三个点，有序构型空间 OC_3 具有由以下单元构成的划分，忽略 z_1、 z_2 和 z_3的置换：

$$\{\mathrm{Re}(z_1) < \mathrm{Re}(z_2) < \mathrm{Re}(z_3)\},\ \{\mathrm{Re}(z_1) < \mathrm{Re}(z_2) = \mathrm{Re}(z_3) \text{ 和 } \mathrm{Im}(z_2) < \mathrm{Im}(z_3)\},\ \{\mathrm{Re}(z_1) = \mathrm{Re}(z_2) < \mathrm{Re}(z_3) \text{ 和 } \mathrm{Im}(z_1) < \mathrm{Im}(z_2)\},\ \{\mathrm{Re}(z_1) = \mathrm{Re}(z_2) = \mathrm{Re}(z_3) \text{ 和 } \mathrm{Im}(z_1) < \mathrm{Im}(z_2) < \mathrm{Im}(z_3)\}。$$

然而，仅凭数值近似（即使是经过认证的近似）来确定包含三元组 $(z_1,\ z_2,\ z_3)$ 的 Fox–Neuwirth胞腔，可能是不可能的，因为无法判定相等。这促使我们改用排列胞腔：它们在 OC_n 中是开集，却能带来丰富得多的组合结构。

排列胞腔的组合多样性，可以通过平面上互不相交的轴对齐矩形（或盒子）所形成的排列来呈现，这类排列也会出现在区间算术中。在此背景下，构型 $(z_1,\ldots,z_n) \in OC_n$ 由每个坐标 $\mathrm{Re}(z_1)$、$\mathrm{Im}(z_1)$、$\mathrm{Re}(z_2)$ 等各自对应的一个区间来描述，因此每个 z_i 都落在一个盒子中，并且我们假设这些盒子两两不相交。两个盒子可以由一条垂直线或一条水平线分隔。因此，一组互不相交的盒子会诱导出一个排列。图 3、4、



图 3. 平面中两个轴对齐互不相交矩形的全部四种组合排列。若对每一对矩形，两种构型中都保持相同的相对位置（即其中一个严格位于另一个的左侧，或严格位于另一个的上方），则这两种构型等价。

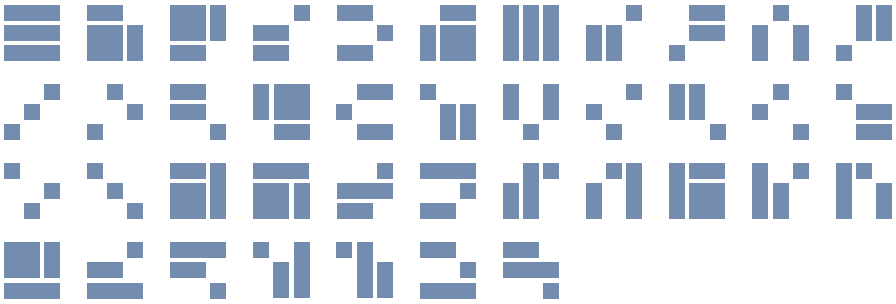


图 4. 平面中三个轴对齐互不相交矩形的全部 40 种组合排列。

和 5 分别展示了这种方式能够得到的所有排列，对应的点数分别为 2、3 和 4。

4. 由排列计算辫子

覆盖路径 $F : [0, 1] \to OC_n$ 的排列胞腔序列完全决定了 F 的同伦类型（引理 6），而算法 2 计算这样的序列。本节聚焦于下一步：计算由 F 诱导的辫子。

4.1. 置换点与基本辫子。 为从覆盖一条路径的排列序列计算辫子，我们使用顶点为置换点的分段线性路径。给定 $[n]$ 的两个置换 π 与 ϕ，置换点 $p(\pi, \phi)$ 是 OC_n 中如下定义的点

$$p(\pi, \phi) = (\pi(1) + \phi(1)\mathbf{i}, \dots, \pi(n) + \phi(n)\mathbf{i}) .$$

注意，对任意置换 $\sigma \in S_n$，$\sigma \cdot p(\pi, \phi) = p(\pi\sigma^{-1}, \phi\sigma^{-1})$。

引理 8. 每个排列胞腔都包含一个置换点。

证明。 论证与引理 4 中证明非空性的方法相同。 □

我们还考虑点 $q(\pi) = (\pi(1) \dots, \pi(n)) \in OC_n$。这些点构成投影 $OC_n \to C_n$ 在基点 $[n] \in C_n$上方的纤维。置换点 $p(\pi, \phi)$ 与 $q(\pi)$ 通过 OC_n中的一条直线路径相连。给定一条连接两个置换点 $p(\pi, \phi)$ 和 $p(\pi', \phi')$ 的路径 P，我们可以将 P延拓为一条从 $q(\pi)$ 到 $q(\pi')$ 的路径 P'：

$$q(\pi) \to p(\pi, \phi) \overset{P}{\to} p(\pi', \phi') \to q(\pi').$$

这表示一条路径：从 $q(\pi)$ 出发，沿直线到达 $p(\pi, \phi)$，沿着 P前往 $p(\pi', \phi')$，然后再沿直线到达 $q(\pi')$。将 P'投影到 C_n上会诱导出一条从基点 $[n]$出发并返回该点的环路，因此其同伦类可用一个组合 $b \in B_n$来描述。反过来，辫子 $b \in B_n$ 是 C_n中一个环路的同伦类。我们可以将这条环路提升为 OC_n 中的一条路径 P'，它从 $q(\pi)$出发，终止于另一点



图 5. 平面中互不相交的 4 个矩形的全部 772 种组合构型。

纤维在 $[n]$ 上的某个 $q(\pi')$。给定两个置换 ϕ 和 ϕ'，我们可以构造扩展路径

$$p(\pi,\phi) \to q(\pi) \overset{P'}{\to} q(\pi') \to p(\pi',\phi'),$$

它在同伦意义下定义良好。我们将这条路径记作 $p(\pi,\ \phi) \overset{b}{\to} p(\pi',\ \phi')$。这两个构造——从 P 到 b 的同伦类，以及从 b、π、ϕ 和 ϕ' 到 P 的同伦类 —— 互为逆构造。此外，它们与复合相容：

$$p(\pi,\phi) \overset{b}{\to} p(\pi',\phi') \overset{b'}{\to} p(\pi'',\phi'') \quad \equiv_{\text{homotopy}} \quad p(\pi,\phi) \overset{b\cdot b'}{\to} p(\pi'',\phi'').$$

给定两个置换 π、ϕ，分段线性路径 $q(\pi) \to p(\pi,\ \phi) \to p(\pi,\ \mathrm{id}) \to q(\pi)$ 位于 OC_n 中，并诱导出平凡辫子：由于实部保持不变，该路径是可缩的。然而，线性路径 $p(\pi,\ \phi) \to p(\mathrm{id}\ \phi)$ 也位于 OC_n 中，却可能诱导出非平凡辫子。令 $B(\pi,\ \phi)$ 表示与之关联的辫子，称为初等辫子。换言之，

$$p(\pi,\phi) \overset{\text{straight line}}{\to} p(\mathrm{id},\phi) \quad \equiv_{\text{homotopy}} \quad p(\pi,\phi) \overset{B(\pi,\phi)}{\to} p(\mathrm{id},\phi).$$

引理 9. 设 A 为一个排列，并设 π、ϕ、π' 和 ϕ' 为 $[n]$ 的置换，使得置换点 $p(\pi,\phi)$ 和 $p(\pi',\phi')$ 位于 A 中。则直线路径 $p(\pi,\phi) \to p(\pi',\phi')$ 包含于 OC_n 中，且其关联辫子为 $B(\pi\pi'^{-1},\ \phi\pi'^{-1})$。

证明. 根据排列胞腔的定义，若 $p(\pi,\phi)$ 与 $p(\pi',\phi')$ 均位于 A 中，则 $p(\pi',\phi)$ 亦然。因此路径 $p(\pi,\phi) \to p(\pi',\phi')$ 同伦于

$$p(\pi,\phi) \to p(\pi',\phi) \to p(\pi',\phi'),$$

由 A 的凸性。在 π'^{-1} 的作用下（该作用不改变关联辫子），此路径变为

$$p(\pi\pi'^{-1},\phi\pi'^{-1}) \to p(\mathrm{id},\phi\pi'^{-1}) \to p(\mathrm{id},\phi'\pi'^{-1}).$$

路径的第一部分按定义诱导辫子 $B(\pi\pi'^{-1},\phi\pi'^{-1})$，而第二部分诱导平凡辫子，因为实部保持不变。□

下面的陈述给出了如何根据 π 分解为初等换位的乘积（$i\ i+1$），来计算 $B(\pi,\phi)$ 的恒等式。

引理 10. 对于任意置换 π, π' 和 ϕ:

(i) $B(id\ \phi) = 1$; *(ii)* $B(\pi'\pi,\phi) = B(\pi',\phi\pi^{-1})B(\pi,\phi)$; *(iii)* $B((i\ i+1)\ \phi) = \sigma_i$，如果 $\phi(i) > \phi(i+1)$，对任意 $1 \le i < n$ 都成立；

(iv) $B((i\ i+1)\ \phi) = \sigma_i^{-1}$，如果 $\phi(i) < \phi(i+1)$，对任意 $1 \le i < n$ 都成立。

证明。 第一点显然，因为线性路径 $p(\mathrm{id}\ \phi) \to p(\mathrm{id}\ \phi)$ 是常值的。

对于第二点，考虑分段直线路径 P

$$p(\pi'\pi,\phi) \to p(\pi,\phi) \to p(\mathrm{id},\phi),$$

以及排列 $A = (\varnothing,\ \prec_\phi)$，其中 $\varnothing$ 为空偏序，而 $\prec_\phi$ 是由 $i \prec_\phi j$ 定义的全序，当 $\phi(i) < \phi(j)$ 时成立。路径 P 完全位于附属于此排列的胞腔内。此胞腔的凸性（引理 4）意味着 P 与直线路径 $p(\pi'\pi,\ \phi) \to p(\mathrm{id}\ \phi)$ 同伦；根据定义，该路径诱导出辫子 $B(\pi'\pi,\ \phi)$。

另一方面，部分 $p(\pi,\ \phi) \to p(\mathrm{id}\ \phi)$ 根据定义诱导出辫子 $B(\pi,\ \phi)$，而部分 $p(\pi'\pi,\ \phi) \to p(\pi,\ \phi)$ 诱导出的辫子与路径诱导出的相同

$$p(\pi',\phi\pi^{-1}) \to p(\mathrm{id},\phi\pi^{-1}),$$

根据置换群作用下的不变性。根据定义，后一条路径诱导出辫子 $B(\pi',\phi\pi^{-1})$。由此得到恒等式 $B(\pi'\pi,\phi) = B(\pi',\phi\pi^{-1})B(\pi,\phi)$。

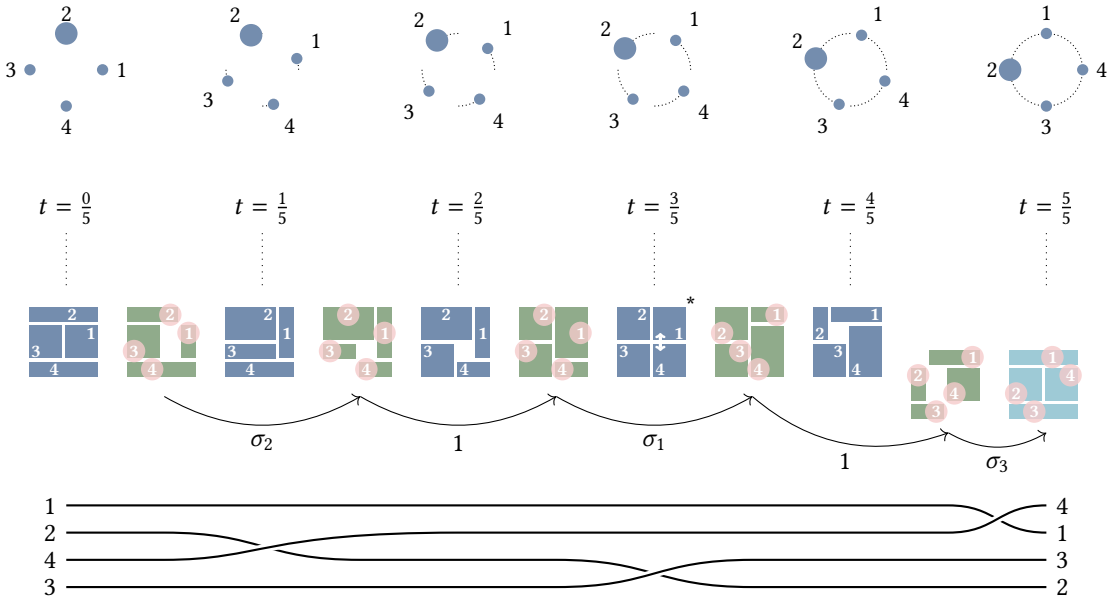


图6. 算法3的示意图：四个点沿圆周移动，续接图 2。蓝色排列图是输入的覆盖序列。绿色排列图表示相邻两个蓝色排列图的交集。粉色圆圈中的数字标出了位于排列胞腔中的一个置换点。最后一个蓝色图表示对覆盖的扩展，使得最后一个排列是第一个排列的置换（见第4.4节）。

对于第三点，令 τ 表示对换 $(i\ i+1)$，并考虑路径 P

$$q(\tau) \to p(\tau, \phi) \to p(\mathrm{id}, \phi) \to q(\mathrm{id}).$$

它被三个排列胞腔覆盖：$(\prec_\tau,\ \varnothing)$、$A$ 和 $(\prec_{\mathrm{id}},\ \varnothing)$，其中 A 是排列 $(\prec_{\mathrm{Re}},\ \prec_{\mathrm{Im}})$，且 $\prec_{\mathrm{Re}} = (\prec_\tau \cap \prec_{\mathrm{id}})$，而 $\prec_{\mathrm{Im}}$ 包含唯一的关系 $i+1 \prec_{\mathrm{Im}} i$。我们很容易验证，这三个排列也覆盖了定义 σ_i 的路径（图1）。最后一个点的情况类似。□

4.2. 由排列序列得到的辫子。 过程 辫子 （算法 3）以排列序列 $A_1, \ldots, A_r$ 作为输入，其中 $A_{i-1} \cap A_i$ 非空。它计算一个置换点序列 $p_1, \ldots, p_r$，使得对 $i \geq 2$ 有 $p_1 \in A_1$ 与 $p_i \in A_{i-1} \cap A_i$。同时，它将由分段线性路径 $p_1 \to \cdots \to p_r$ 诱导的辫子 b 计算为基本辫子 $B(\pi, \phi)$ 的复合（第 4.1 节）。最后，它输出三元组 (p_1, b, p_r)。

图 6 给出了一个例子，延续图 2 中的例子。在此情形下，该算法（在若干可能选择中）计算出如下置换点：

$$p_1 = p\,(4\,3\,1\,2, 3\,4\,2\,1) \qquad p_2 = p\,(4\,2\,1\,3, 3\,4\,2\,1)$$
$$p_3 = p_2 \qquad p_4 = p\,(4\,1\,2\,3, 4\,3\,2\,1),$$

其中，例如，4 3 1 2 表示置换 $\{1 \mapsto 4, 2 \mapsto 3, 3 \mapsto 1, 4 \mapsto 2\}$；并输出三元组 $(p_1, \sigma_2\sigma_1, p_4)$。

定理 11 (辫子的正确性)。 设 $(A_1, \ldots, A_r)$ 是覆盖路径 $F : [0,1] \to OC_n$ 的一列排列。设 (p, b, p') 是算法 *3* 在输入 $A_1, \ldots, A_r$ 上的输出三元组。那么：

- $F(0)$ 和 p 属于 A_1；
- $F(1)$ 和 p' 位于 A_r 中；

算法 3 计算与排列序列关联的辫子

输入： 一系列排列。
输出： 一个三元组 (p, b, p')，其中 p 和 p' 是置换点，而 b 是一条辫子。

定义 辫子$(A_1, \dots, A_r)$**:**
  $(\pi_{\text{start}}, \phi_{\text{start}}) \leftarrow$ 是 A_1中的一个置换点
  $(\pi, \phi) \leftarrow (\pi_{\text{start}}, \phi_{\text{start}})$
  $b \leftarrow 1$ # 平凡辫子
  对于 i，**从** 2 **到** r**:**
    如果 $A_{i-1} \cap A_i = \varnothing$**:**
      错误
    $(\pi', \phi') \leftarrow$ 在 $A_{i-1} \cap A_i$ 中的一个置换点
    $b \leftarrow b \cdot B(\pi\pi'^{-1}, \phi\pi'^{-1})$
    $(\pi, \phi) \leftarrow (\pi', \phi')$
  返回 $p(\pi_{\text{start}}, \phi_{\text{start}})$ b，$p(\pi, \phi)$

- *the path* $F(0) \to p \xrightarrow{b} p' \to F(1)$ *lies in* OC_n *and is homotopic to* F.

证明。 根据覆盖的定义（定义 5），交集 $A_{i-1} \cap A_i$ 非空，因此根据引理 4 和引理 8，每个交集都包含一个置换点。所以算法不会失败。前两个点由构造可得。

设 $p_1 = p(\pi_{\text{start}}, \phi_{\text{start}})$，并令 p_i为在 $A_{i-1} \cap A_i$中选取的置换点。显然，分段线性路径 $p_1 \to \cdots \to p_r$也被 A_1、……、A_r覆盖。根据反复应用引理 9，其关联辫子为 b。由于 $F(0) \in A_1$ 且 $F(1) \in A_r$，分段线性路径

$$F(0) \to p_1 \to \cdots \to p_r \to F(1)$$

也被同一序列覆盖。根据引理 6，该路径与 F同伦。 □

4.3. 路径的复合。 给定两条路径 $F, G : [0, 1] \to OC_n$，其中 $F(1) = G(0)$，目标是使用上述算法计算与路径拼接 $F \cdot G$ 对应的辫子。对 F和 G 应用 覆盖 （算法 2），得到排列的覆盖序列（$A_1, \dots, A_r$）和（$B_1, \dots, B_s$）。它们的拼接

$$A_1, \dots, A_r, B_1, \dots, B_s$$

覆盖了 $F \cdot G$，因此算法 3 可直接生成所需的辫子。不过，最好只存储 辫子 $(A_1, \dots, A_r)$ 和 辫子 $(B_1, \dots, B_s)$ 的输出；记它们为

$$\big((\pi_F, \phi_F), b_F, (\pi'_F, \phi'_F)\big) \text{ and } \big((\pi_G, \phi_G), b_G, (\pi'_G, \phi'_G)\big).$$

当一个排列胞腔 A同时包含 $p(\pi'_F, \phi'_F)$ 和 $p(\pi_G, \phi_G)$，以及共同端点 $F(1) = G(0)$ 时，路径的复合

$$p(\pi_F, \phi_F) \xrightarrow{b_F} p(\pi'_F, \phi'_F) \to F(1) = G(0) \to p(\pi_G, \phi_G) \xrightarrow{b_G} p(\pi'_G, \phi'_G) \tag{1}$$

根据 A的凸性，同伦于

$$p(\pi_F, \phi_F) \xrightarrow{b_F} p(\pi'_F, \phi'_F) \to p(\pi_G, \phi_G) \xrightarrow{b_G} p(\pi'_G, \phi'_G),$$

其辫子等于

$$b_F \cdot B(\pi'_F \pi_G^{-1}, \phi'_F \pi_G^{-1}) \cdot b_G.$$

然而，仅有等式 $F(1) = G(0)$ 并不能确保 $p(\pi'_F, \phi'_F)$ 和 $p(\pi_G, \phi_G)$ 位于同一个排列胞腔中；连接它们的直线段甚至可能离开 OC_n。我们建议采用以下两种方法来规避这一问题：

（1）当未来的复合已预先知晓时，将 B_1 添加到 $(A_1$、……、$A_r)$ 中。扩展后的序列仍然覆盖 F，这将确保 $F(1)$、$p(\pi'_F,\ \phi'_F)$ 和 $p(\pi_G,\ \phi_G)$ 位于公共单元 B_1 中。

(2) 保留端点胞腔 A_r 和 B_1 并计算它们的交集。该交集非空，因为其中包含 $F(1) = G(0)$。选择一个置换点 $p(\pi,\ \phi)$，使其属于 $A_r \cap B_1$；那么路径 (1) 同伦于

$$p(\pi_F,\phi_F) \overset{b_F}{\to} p(\pi'_F,\phi'_F) \to p(\pi,\phi) \to p(\pi_G,\phi_G) \overset{b_G}{\to} p(\pi'_G,\phi'_G)$$

其关联辫子为

$$b_F \cdot B(\pi'_F\pi^{-1}, \phi'_F\pi^{-1}) \cdot B(\pi\pi_G^{-1}, \phi\pi_G^{-1}) \cdot b_G.$$

接下来，我们放宽条件 $F(1) = G(0)$，只要求在 C_n 中相等，即存在置换 σ 使得 $\sigma \cdot F(1) = G(0)$。我们考虑由 F 和 G 分别在 C_n 中诱导的路径 F 和 G，因此 $F(1) = G(0)$。如何根据为 F 和 G 计算出的辫子数据，计算复合 $F \cdot G$ 由~ ~诱导的辫子？若已知置换 σ，则问题可化为前一种情况 $F(1) = G(0)$。遗憾的是，分离方法 *sep* 不足以计算 σ。例如，对于两个点，假设某个 σ（它是恒等置换或对换 (1 2)）满足 $F(1) = \sigma \cdot G(0)$，并且

$$sep(F, 1, 2, 1) = (2, 1, \mathrm{Re}, [\dots]) \text{ and } sep(G, 1, 2, 0) = (1, 2, \mathrm{Im}, [\dots]),$$

这意味着 $\mathrm{Re}(F_1(1)) > \mathrm{Re}(F_2(1))$ 且 $\mathrm{Im}(G_1(0)) < \mathrm{Im}(G_2(0))$。这些信息足以恢复 σ 吗？答案是否定的：可能是 $\sigma = \mathrm{id}$，其中 $F(1) = G(0) = (1,\ \mathrm{i})$；也可能是 $\sigma = (1\ 2)$，其中 $F(1) = (1 + \mathrm{i},\ 0)$，且 $G(0) = (0,\ 1 + \mathrm{i})$。因此，我们需要通过其他方法（例如路径跟踪）来计算 σ。

4.4. 沿环路的辫子。考虑一条路径 $F : [0, 1] \to OC_n$，它在 C_n 中诱导一个环路。换言之，存在一个置换 σ，使得 $F(0) = \sigma \cdot F(1)$。我们要寻找一个置换点 $p(\pi,\ \phi)$ 和一个辫子 b，使得

$$F(0) \to p(\pi,\phi) \overset{b}{\to} \sigma^{-1} \cdot p(\pi,\phi) \to F(1)$$

与 F 同伦。

这个问题与上面的复合问题类似。具体而言，我们必须通过其他方法确定 σ，因为单靠 *sep* 方法无法提供足够的信息来恢复它。

一旦已知 σ，我们就计算 F 的一个覆盖 A_1、……、A_r，并将其扩展为

$$A_1, \dots, A_r, \sigma^{-1}A_1$$

由于 $F(0) = \sigma \cdot F(1)$ 且 $F(0) \in A_1$，扩展后的序列仍覆盖 F。方法辫子 $(A_1$、……、A_r、$\sigma^{-1}A_1)$ 输出一个三元组 $(\pi$、$\phi)$、b 和 $(\pi'$、$\phi')$，使得

$$F(0) \to p(\pi,\phi) \overset{b}{\to} p(\pi',\phi') \to F(1)$$

与 F 同伦。此外，$F(0) = \sigma \cdot F(1) \in A_1$，$p(\pi$、$\phi) \in A_1$ 且 $p(\pi'$、$\phi') \in \sigma^{-1}A_1$。因此，上述路径与

$$F(0) \to p(\pi,\phi) \overset{b}{\to} p(\pi',\phi') \to \sigma^{-1} \cdot p(\pi,\phi) \to F(1),$$

由 $\sigma^{-1}A_1$的凸性可知，并且它可简化为

$$F(0) \to p(\pi,\phi) \overset{\tilde{b}}{\to} \sigma^{-1} \cdot p(\pi,\phi) \to F(1),$$

其中 $\tilde{b} = b \cdot B(\pi'\sigma^{-1}\pi^{-1}$、$\phi'\sigma^{-1}\pi^{-1})$，根据引理 9。

假设我们已经计算出 辫子$(A_1$、……、$A_r)$，并保留了 A_1 和 A_r。与路径复合问题类似，我们可以计算交集 $A_r \cap \sigma^{-1}A_1$ 中的一个置换点，并利用它闭合环路（见图 6）。

5. 实现

前几节介绍的用于计算由路径 $F : [0,1] \to OC_n$ 诱导的辫子的算法分为两步：首先计算覆盖 F 的排列序列（算法 2）；然后在序列中每一对相邻排列的交集中找到一个置换点，并即时计算与这些置换点的直线插值相关联的辫子（算法 3）。

本节从实现的角度重新审视算法 2 和 3。在算法 2 第 13 行，工作排列会被复制，从而生成一个排列列表，供算法 3 后续处理。为避免复制并简化实现，我们将算法 2 和 3 合并：维护一个当前排列 A 和一个置换点 p（位于 A中）；每当 A 更新时，就更新 p，使其同时位于旧的 A和新的 A中。文中给出了排列和置换点的显式数据结构，并详细描述了这一更新过程（第 5.2 节）。最后，重新考察算法 2 第 8 行的环路：删除一条边后，需要找到不可比较对（i, j）（第 5.3 节）。总之，这样便得到了一些简单的算法，可用基本数据结构实现。

5.1. 带基点排列的数据结构。 回顾一下，排列是定义在 $[n]$ 上的一对偏序 $(\prec_{\mathrm{Re}}, \prec_{\mathrm{Im}})$，其中所有 $i \neq j$ 在 $\prec_{\mathrm{Re}}$ 或 $\prec_{\mathrm{Im}}$ 中都是可比较的（见第 3.2 节）。

定义在 $[n]$ 上的偏序 $\prec$ 用一个顶点为 $[n]$ 的有向无环图（DAG）G 编码，其中 $i \prec j$ 成立，当且仅当在 G 中存在一条从 i 到 j 的有向路径。排列 $(\prec_{\mathrm{Re}}, \prec_{\mathrm{Im}})$ 用一个图 G 编码，其边标记为 Re 或 Im，因此其边集是 $[n]^2 \times \{\mathrm{Re}, \mathrm{Im}\}$ 的子集。sep 的一个输出 (i, j, q, t) 对应于一条边 (i, j, q)，其生存期为 t。令 G_{Re}（分别为 G_{Im}）表示 G 中由标记为 Re（分别为 Im）的边构成的子图；当 (i, j) 是 G_{Re}（分别为 G_{Im}）中的一条边时，记作 $i \to_{\mathrm{Re}} j$（分别为 $i \to_{\mathrm{Im}} j$）。借代地，表示 $(\prec_{\mathrm{Re}}, \prec_{\mathrm{Im}})$ 的这种图称为排列，如果 $(\prec_{\mathrm{Re}}, \prec_{\mathrm{Im}})$ 是排列的话。

令 $(\prec_{\mathrm{Re}}, \prec_{\mathrm{Im}})$ 为与 G 相关联的偏序对，A 为相应的胞腔，定义见第 3.2 节。令 $\pi, \phi \in S_n$。置换点 $p(\pi, \phi)$ 属于 A，当且仅当 π 沿着 G_{Re} 的边递增，且 ϕ 沿着 G_{Im} 的边递增；也就是说，对于所有 $i, j \in [n]$,

$$i \to_{\mathrm{Re}} j \Rightarrow \pi(i) < \pi(j) \quad \text{and} \quad i \to_{\mathrm{Im}} j \Rightarrow \phi(i) < \phi(j). \tag{2}$$

若此条件成立，我们记作 $G \vdash p(\pi, \phi)$。换言之，将 $[n]$ 的置换视为 $[n]$ 上的全序，则 $G \vdash p(\pi, \phi)$ 当且仅当置换 π 和 ϕ 分别是 G_{Re} 和 G_{Im} 的拓扑序。

在计算辫子时，我们会维护一个排列，使一个置换点位于其相应的胞腔内；这就是一个带基点排列。

5.2. 带基点排列的更新。 带基点排列有两种操作：在 G 中删除一条边和插入一条边，同时保持条件 $G \vdash p(\pi, \phi)$。将后一条件理解为 π 和 ϕ 分别是 G_{Re} 和 G_{Im} 上的拓扑序，这就是动态拓扑排序问题。若从 G' 中删除一条边得到 G，那么只要 $G \vdash p(\pi, \phi)$，就有 $G' \vdash p(\pi, \phi)$；因此，从 G 中删除一条边时，π 和 ϕ 可以保持不变。另一方面，在 G 中插入一条边可能需要更新 π 和 ϕ：这正是插入所执行的操作（算法 4）。由于 G 并未完全改变，更新 π 和 ϕ 比从头重新计算更合适。为此，我们改编了 Marchetti-Spaccamela 等人（1996）的算法（另见 Pearce 和 Kelly，2007）（图 7）。该算法会原地修改 G, π 和 ϕ，因此它们是伪代码中的全局变量。

我们考察一次插入（算法 4）的运行。设 G'，π'，ϕ' 为运行插入之前带基点排列的状态，且 $(i, j, q) \in [n]^2 \times \{\mathrm{Re}, \mathrm{Im}\}$ 是一条有效的边

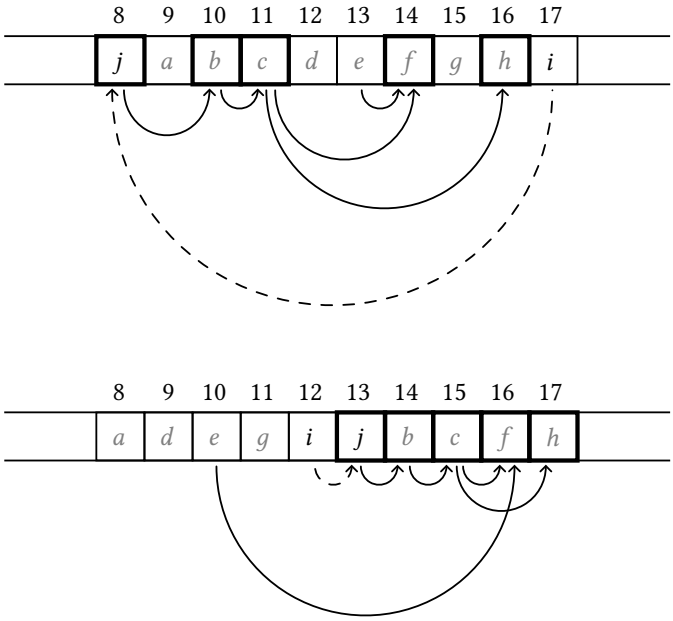


图 7. 更新实数$(i,\ j)$的示意图。我们将 π^{-1}绘制为一个水平数组，把索引映射到顶点（例如，$\pi(j)\ =8$），并将重新标记的边绘制在 G中。若所有边都指向右侧，则拓扑排序正确。从 i 指向 j的虚线箭头表示待插入的边。使用算法 4中的变量名，$m=8$、$M=17$和 δ是对应于粗体方块的顶点集合。顶点 i不能属于 δ，因为假设新边不会引入环。区间 $[m,M]$内的顶点会重新排序，将 δ中的元素推向右侧，其余元素推向左侧。完成后，从 i指向 j的虚线箭头会指向右侧，因为 i不在 δ中。该过程始终使已有边指向右侧。尽管存在更高效的实现方法（Pearce 和 Kelly，2007年），但必须使用初等换位更新 π，才能用标准生成元表示辫子。

，其中 G'，这意味着将 $(i,\ j,\ q)$ 插入 G'不会在 G'_{Re} 中引入任何环，也不会在 G'_{Im} 中引入任何环。设 $G,\ \pi,\ \phi$为运行插入$(i,\ j,\ q)$ 后排列的状态，b为输出。

命题 12. 假设初始图 G' 是一个排列，则算法 *4* 正确：使用上述记号，$G=G'\cup\{(i,j,q)\}$、$G\vdash p(\pi,\phi)$以及 b是由定义良好的分段直线路径 $p(\pi',\phi')\ \to\ p(\pi,\phi)$所诱导的辫子。

证明。 当 $q=\mathrm{Im}$ 时，过程 *insert* 调用 *update imag*，它会更新拓扑排序 ϕ，使其与新边兼容，并返回平凡辫子。事实上，路径 $p(\pi',\phi')\ \to p(\pi,\phi)$ 诱导恒等辫子，因为 $\pi=\pi'$。

当 $q\ =\mathrm{Re}$ 时，过程 *insert* 调用 *update real*，它同时执行两项任务。首先，它遵循 Marchetti-Spaccamela 等人（1996年）的策略（见图 7），通过从左侧乘以合适的初等换位（第 12 行）来更新拓扑排序 π，使其与新边兼容。其次，它计算由 $p(\pi',\phi)\ \to p(\pi,\phi)$ 诱导的辫子，其中 π'是 π的初始值。

我们声称，在该过程结束时，b是由 $p(\pi',\ \phi)\ \to p(\pi,\ \phi)$ 诱导的辫子。根据拓扑排序算法的构造，在整个过程中，$G'\vdash p(\pi,\ \phi)$，因此路径 $p(\pi',\ \phi)\ \to p(\pi,\ \phi)$ 位于由 G'定义的排列胞腔中，故引理 9 适用。该路径诱导辫子 $B(\pi'\pi^{-1},\ \phi\pi^{-1})$。置换 $\pi'\pi^{-1}$ 分解为

算法 4 带基点排列中的边插入

```
1  定义 更新实数(i, j):  # 更新 π在插入 (i, j,实部) 在 G
2      m ← π(j);   M ← π(i);   c ← 1;   b ← 1
3      δ ← {k ∈ [n] : j <_Re k且 π(k) < M}# 有界深度优先搜索
4      对于 x从 m + 1 到 M:
5          如果 π^{-1}(x) ∈ δ:
6              c ← c + 1;
7              继续
8          对于 y从 x到 x − c + 1:# 逆序迭代
9              如果 φ(π^{-1}(y − 1)) < φ(π^{-1}(y)):
10                 b ← b · σ_{y−1}
11             否则:
12                 b ← b · σ_{y−1}^{-1}
               π ← (y − 1 y)π
13     返回 b
14
15 定义 更新图像(i,  j):  # 更新 φ在插入 (i,  j,  Im) 到 G
16     使用动态拓扑排序算法更新 φ
17     返回 1
18
19 定义插入(i, j, q):
20     G ← G ∪ {(i,  j,  q)}
21     如果 q = 实部:
22         返回更新实数(i, j)
23     否则:
24         返回更新 imag(i, j)
```

一系列在第 12 行累积的初等换位的乘积，并使用引理 10 计算辫子：每个对换 $(y-1\ y)$ 都会为 b 贡献一个因子 σ_{y-1} 或 σ_{y-1}^{-1}，这正是第 8 行所计算的内容。□

5.3. **辫子计算实现**。接下来，算法 2 和算法 3 被合并为算法 5 的 *braid'* 函数。它依赖算法 4 在当前排列中维护一个置换点，并即时计算辫子。

过程 *braid'*（算法 5）首先计算一个包含 $F(0)$ 的初始排列 G，以及该排列中的一个置换点。G 的每条边都关联一个生存期。当所有生存期均至少为 1 时，算法终止。

定理 13（*braid'*的正确性）。对于以近似形式给出的输入 $F : [0,1] \to OC_n$，算法 5 输出一个三元组 $((G_0, \pi_0, \phi_0)b, (G_1, \pi_1, \phi_1))$，使得

- G_0 是一个包含 $p(\pi_0, \phi_0)$ 和 $F(0)$的排列；
- G_1 是一个包含 $p(\pi_1, \phi_1)$ 和 $F(1)$的排列；
- 路径 $F(0) \to p(\pi_0, \phi_0) \xrightarrow{b} p(\pi_1, \phi_1) \to F(1)$ 位于 OC_n 中，且与 F同伦。

5.3.1. 基本不变量。我们证明主循环保持以下不变量：G是一个排列，且 $G \vdash p(\pi,\ \phi)$。

在主循环的每次迭代中，选取 G 中生存期最短的边进行删除，例如 $(i,\ j,\ q)$。删除该边可能会破坏 G 是一个排列这一不变量

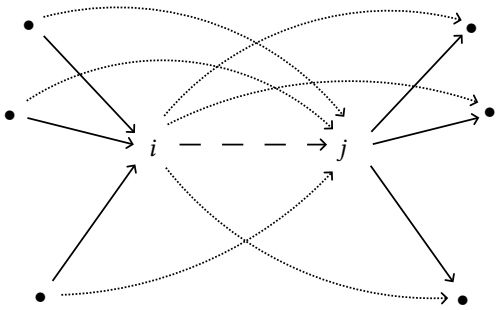


图 8。准备删除边的过程。我们假设要删除的边是 $i \to_{\mathrm{Re}} j$，并且只表示 Re-标记的边。初始时，只有普通边和从 i到 j的边存在于 G_{Re}中。在 G_{Re}中插入虚线边、在 G_{Im}中插入边 $i \to j$ 或 $j \to i$，并删除边 $i \to_{\mathrm{Re}} j$之后，图 G仍然是一个排列。

并产生不可比较对。如果存在一条从 i 到 j、带有 q 标记的路径，该路径位于 $G \setminus \{(i,\ j,\ q)\}$ 中，那么删除该边不会产生不可比较对。否则，该过程会插入新边，使删除操作得以进行（第 17 行之后的代码块）。

为此，该过程会查询*sep*（第 18 行），它输出一条新的边（i', j', q'），其生存期为 t'。如果 $q = q'$（第 19 行），那么 $i = i'$ 和 $j = j'$，因为否则 $i = j'$ 和 $j = i'$，这意味着 $q(F_i(t)) < q(F_j(t))$ 和 $q(F_i(t)) > q(F_j(t))$ 同时成立。因此在这种情况下，该过程不会删除这条边，而是将其生存期延长至 t'，然后继续下一次迭代。

在 $q \neq q'$（第 22 行）的情况下，使用插入过程插入新边，同时更新 π和 ϕ，并即时计算辫子。但已过期的边（i, j, q）尚未满足删除条件：$G \setminus \{(i, j, q)\}$中仍可能存在不可比较对。假设确实存在这样一对$\{u, v\}$，这意味着在 G中，存在一条从 u到 v（或反向）的 Re-或Im-标记路径，但在 $G \setminus \{(i, j, q)\}$中不存在。它不可能是$\{i, j\}$，因为新插入的边（$i', j', q'$）使 i和 j变得可比较。为解决这一问题，假设在 G中存在一条从 u到 v的 Re-或Im-标记路径，而 $v \neq j$（$u \neq i$的情况类似）。这条路径在 $G \setminus \{(i, j, q)\}$中不存在，因此它带有 q标记，并经过边 $i \to_q j$。由于 $j \neq v$，从 u到 v的路径中，边 $i \to_q j$之后必定有一条边 $j \to_q k$。该过程会准确地插入所有边 $i \to_q k$（每当 $j \to_q k$时）和 $k \to_q i$（每当 $k \to_q j$时），见图 8。因此，在第 32 行之后，从 u到 v的路径中的线段 $i \to_q j \to_q k$可以替换为边 $i \to_q k$，从而得到一条在 $G \setminus \{(i, j, q)\}$中从 u到 b的路径。因此，$G \setminus \{(i, j, q)\}$中不存在不可比较对。请注意，可以安全地插入边 $i \to_q k$和 $k \to_q i$，其生存期为 t，因为它们不会改变关系$<_q$。

5.3.2. 终止性。令 $\varepsilon > 0$ 由条件（*sep*$_\varepsilon$）给出。在每次迭代中，环路都会选择生存期最短的一条边 t。对这条边调用*sep*时，返回的生存期满足 $t' \geq \min(1, t + \varepsilon)$，因此，每次调用都会使相应关系的生存期至少增加 ε，直至达到 1。由于有序对和标签的数量有限，这种情况只能发生有限次。

插入不会增加生存期的额外边时（第 22–32 行），G_q中经过新边（i, k, q）或（k, j, q）的路径数量，严格少于经过（i, j, q）的路径数量；后者会被删除。这确保主循环不会停滞在给定的 t。

5.3.3. 正确性。接下来还需验证 *braid'*确实计算出了我们想要的结果。设 G_k、π_k、ϕ_k 分别为第 k次迭代结束时相应变量的值。考虑

第33行之后各变量的状态，基本不变量（第5.3.1节）告诉我们，G_k包含 $p(\pi_k,\ \phi_k)$。考虑第33行之前各变量的状态，它告诉我们，G_{k-1}也包含 $p(\pi_k,\ \phi_k)$。此外，由于新边是按照 *sep*插入的，或者是利用偏序的传递性插入的（第27–32行），因此排列序列 (G_k) 覆盖 F。

根据命题 12，在第 k 次迭代中，$insert(i',\ j',\ q')$ 返回由路径 $p(\pi_{k-1},\ \phi_{k-1}) \to p(\pi_k,\ \phi_k)$ 诱导的辫子。不调用 *insert*的迭代不会修改 π 和 ϕ。因此，正确性与算法 3（定理 11）的正确性相同。

5.3.4. 排列初始化。 下面说明初始排列 G 的计算方法（第5–11行）。对于每个不同的 $i,\ j \in [n]$，我们使用 *sep* 计算 i 与 j 之间的一条边，从而在 n 中产生二次复杂度。这确保了 G 是一个排列，但其中可能存在冗余边。根据 *sep* 的具体实现以及对 $F(0)$ 的附加条件，可以改进此初始化过程。例如，如果 *sep* 在时刻0仅返回 Re-标记的边，则可以通过使用 *sep* 按实部对 $F(0)$ 的分量进行排序来初始化排列。对于该排列 G，G_{Im} 为空，而 G_{Re} 包含形如 $(i,\ i+1,\ \mathrm{Re})$ 的边（对于 $i \in [n-1]$），它将包含 $F(0)$。这一初始化过程关于 n 是准线性的，而朴素方法具有二次复杂度。当 *sep* 大多数时候仅返回 Re-标记的边时，也可以采用这一策略。此外，我们可以利用关于 $F(0)$ 的先验信息，直接计算一个包含它的排列，例如 $F(0)$ 的分量已经按字典序排序。

5.4. **sep 的实际实现*sep***。我们的主要动机是计算参数化多项式 $f \in \mathbb{C}[t][x]$ 的根在位移下定义的辫子，其中 t沿着一条避开 f 的判别式簇的连续路径 $\gamma : [0,1] \to \mathbb{C}$ 移动。要使用上文介绍的方法计算辫子，我们需要近似由 $f(\gamma(t)\ \text{–})$ 的根在 OC_n 中形成的路径，具体含义见定义 2。

经认证的同伦延拓软件（例如 Beltr´an 和 Leykin，2013；Duff 和 Lee，2024；Guillemot 和 Lairez，2024）以 f和 γ 为输入，并返回围绕根路径的互不相交的管状邻域。具体来说，我们考虑 *algpath*[4]（Guillemot & Lairez，2024），它依赖于区间算术。

设 $F : [0,1] \to OC_n$为一条路径，当 t变化时，它描述 $f(\gamma(t)\ \text{–})$ 的根。用 $\square[0,1]$ 表示 $[0,1]$, 的闭子区间，用 $\square\mathbb{C}$ 表示实紧区间对（即平面中的盒子）的集合。对于每个 $i \in [n]$，*algpath* 都会返回 F_i 的一个区间扩张 $\square F_i : \square[0,1] \to \square\mathbb{C}$。它们由泰勒模型序列表示（Guillemot & Lairez，2024，第 6.2 节）。我们可以确保，对于所有 $t \in [0,1]$ 以及互不相同的 $i,\ j \in [n]$,

(1) $\square F_i(\{t\}) \cap \square F_j(\{t\}) = \varnothing$;

(2) $\square F_i$ 在豪斯多夫拓扑下于 $\{t\}$ 连续。

过程 *sep* 的实现如下。令 $i, j \in [n]$ 互不相同且 $t \in [0,1)$。设 $\delta = 1 - t$，则

(1) 计算 $X_i = \square F_i([t, t+\delta]) \subseteq \mathbb{C}$ 和 $X_j = \square F_j([t, t+\delta]) \subseteq \mathbb{C}$；(2) ● 若 $\mathrm{Re}\ X_i < \mathrm{Re}\ X_j$，返回 $(i, j, \mathrm{Re}\ t+\delta)$；

- ● 若 $\mathrm{Re}\ X_i > \mathrm{Re}\ X_j$，返回 $(j, i, \mathrm{Re}\ t+\delta)$;
- ● 若 $\mathrm{Im}\ X_i < \mathrm{Im}\ X_j$，返回 $(i, j, \mathrm{Im}\ t+\delta)$;
- ● 如果 $\mathrm{Im}\ X_i > \mathrm{Im}\ X_j$，则返回 $(j, i, \mathrm{Im}\ t+\delta)$;

(3) 否则将 δ 除以 2，然后返回步骤 1。

由于 $\square F_i(\{t\})$ 和 $\square F_j(\{t\})$ 不相交，$\square F_i$ 和 $\square F_j$ 在 $\{t\}$ 处的连续性保证了当 δ 足够小时，$\square F_i([t, t+\delta])$ 和 $\square F_j([t, t+\delta])$ 不相交。后两者是复区间，因此步骤2中的某个条件必然成立，过程随即终止。至于定义1，条目（sep_a）和（sep_c）显然成立，而条目（sep_b）则由 $[0,1]$ 的紧致性以及条件1和2推出。

[4]https://gitlab.inria.fr/numag/algpath

本文所述辫子计算算法的 Rust 实现可在 https://gitlab.inria.fr/aguillem/braids 获取。它提供了一个用于计算辫子单值化的命令行界面，并以 *algpath* 作为经认证的同伦延拓后端，其中已按照上文所述实现了 *sep* 。

参考文献

Artin, E. (1947年). 辫子理论. 《数学年刊》, *48*(1), 101–126. https://doi.org/10.2307/1969218 Beltr´an, C., & Leykin, A. (2012). 经认证的数值同伦跟踪. 《实验数学》, *21*(1), 69–83. https://doi.org/10.1080/10586458.2011.606184 Beltr´an, C., & Leykin, A. (2013). 稳健的经认证数值同伦跟踪. 《计算数学基础》, *13*(2), 253–295. https://doi.org/10.1007/S10208-013-9143-2 Duff, T., & Lee, K. (2024年). 使用 Krawczyk 方法的经认证同伦跟踪. *ISSAC 2024* 论文集. Fox, R., & Neuwirth, L. (1962). 辫子群. 《斯堪的纳维亚数学杂志》, *10*, 119–126. 检索于 2024年，来源：https://www.jstor.org/stable/24489274 Guillemot, A., & Lairez, P. (2024*2024*年, 36 45 10 1145 3666000 3669673). 代数路径跟踪的验证数值计算. *ISSAC* 论文集 – . https://doi.org/ . / . Hatcher, A. (2002). 代数拓扑. 剑桥大学出版社. Kranich, S. (2016 年). 平面代数曲线的一个 ε-δ 界及其在平面代数曲线系统经认证同伦延拓中的应用. https://doi.org/10.48550/ arXiv.1505.03432 Marchetti-Spaccamela, A., Nanni, U., & Rohnert, H. (1996年). 在边插入时维护拓扑序. 《信息处理快报》, *59*(1), 53–58. https://doi.org/10.1016/0020-0190(96)00075-0 Marco-Buzunariz, M. ´A., & Rodr´ıguez, M. (2016年). SIROCCO：用于经认证的多项式根延拓的库. *ICMS 2016* 论文集, 191–197. https://doi.org/10.1007/978-3-319-42432-3 24 Pearce, D. J., & Kelly, P. H. J. (2007年). 用于有向无环图的动态拓扑排序算法. 《*ACM* 实验算法学杂志》, *11*, 1.7–es. https://doi.org/10.1145/1187436.1210590 Pichon-Pharabod, E. (2024年). 代数几何中的周期：费曼积分的计算与应用(出版物编号 2024UPASM034) [博士论文，巴黎-萨克雷大学]。 https://theses.hal.science/tel-04823423 Rodriguez, J. I., & Wang, B. (2017年). 辫子群的数值计算. arXiv: 1711.07947. van der Hoeven, J. (2011年). 可靠的同伦延拓. https://hal.science/hal-00589948v4 Van Kampen, E. R. (1933). 关于代数曲线的基本群. 《美国数学杂志》, *55*(1), 255–260. https://doi.org/10.2307/2371128 Xu, J., Burr, M., & Yap, C. (2018). 一种认证同伦延拓路径的方法：单变量情形. *ISSAC 2018* 论文集, 399–406. https://doi.org/10.1145/3208976.3209010 Zariski, O. (1929). 关于存在具有给定分支曲线的二元代数函数的问题. 《美国数学杂志》, *51*(2), 305–328. https://doi.org/10.2307/2370712

Inria ，大学 巴黎–萨克雷 ，帕莱索 , 91120, 法国
电子邮箱：alexandre.guillemot@inria.fr

Inria，巴黎萨克雷大学，帕莱索，91120，法国电子邮件
地址：pierre.lairez@inria.fr

算法 5 辫子

```
def braid'(F):
    G ←{}
    生存期 ← {} # 将 G 的边映射到其生存期
    b ← 1
    对于 i，在 1 , . . . , n:
        对于 j在 i + 1中, . . . , n:
            e, t ← sep（F, i, j, 0）
            G ← G ∪ {e}
            生存期[e] ← t
    π ← 拓扑排序(G_Re);          φ ← 拓扑排序(G_Im)
    G_0, π_0, φ_0 ← π, φ # 保存初始带基点排列

    环路:
        (i, j, q) ← G 中生存期最短的边
        如果生存期[i, j, q] ≥ 1:
            跳出
        如果 在 G ∖ {(i, j, q)} 中不存在从 i 到 j 的以 q 标记的路径:
            i′, j′, q′, t′ ← sep（F, i, j, t）
            如果 q = q′: # 必然地， i = i′ 并且 j = j′
                生存期[i, j, q] ← t′
                继续
            否则:
                # 将边（i′， j′， q′）插入 G，并更新拓扑序 π 和 φ。
                b ← 插入（i′， j′， q′） · b
                生存期[i′, j′, q′] ← t′
                # 插入额外的边以准备删除
                对于 k ∈ [n]，使得 k →_q i 且非 k →_q j:
                    G ← G ∪ {(k, j, q)}
                    生存期[k, j, q] ← t
                对于 k ∈ [n]，使得 j →_q k 且非 i →_q k:
                    G ← G ∪ {(i, k, q)};
                    生存期[i, j, q] ← t
        G ← G ∖ {(i， j， q)}

    返回 (G_0, π_0, φ_0) b, (G_1, π, φ)
```

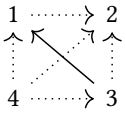
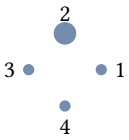
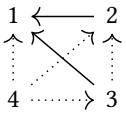
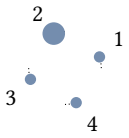
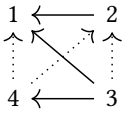
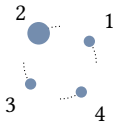
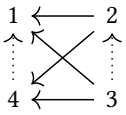
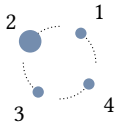
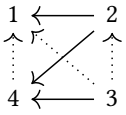
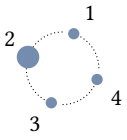
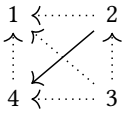
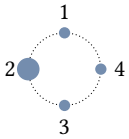
t	G	π^{-1}	ϕ^{-1}	b	$F(t)$
0		3 4 2 1	4 3 1 2	1	
$\frac{1}{5}$		3 4 2 1	4 3 1 2	1	
$\frac{2}{5}$		3 4 2 1	4 3 1 2	1	
$\frac{3}{5}$		3 2 4 1	4 3 1 2	σ_2	
$\frac{4}{5}$		3 2 4 1	4 3 1 2	σ_2	
1		3 2 4 1 2 3 1 4	3 4 2 1 3 2 4 1	σ_2 $\sigma_2\sigma_1\sigma_3$	

图 9. 图 2 所述路径的辫子计算。 t： $[0,1]$中的时间。 G：由图表示的排列。普通边对应 Re-标记的边，虚线边对应虚部标记边。 π^{-1}， ϕ^{-1}： $[4]$上的置换，它们由各自取值构成的数组表示一个置换点。 b：4 股辫子。 F：路径在 t时刻的 F。表格上半部分记录了算法 5 中这些变量在初始化之后以及每次迭代结束时的值。列 F表明， $F(0) = \sigma \cdot F(1)$，其中 $\sigma = 2\,3\,4\,1$。在下半部分，我们按照 4.4 节所述，通过将 $\sigma^{-1} G_0$加入排列序列来闭合环路。将 $t = 0$处的置换记为 π_0， ϕ_0（将 $t = 1$处的置换记为 π_1， ϕ_1）。最后，将 b与$\sigma_1\sigma_3$相乘，完成计算；$\sigma_1\sigma_3$是由直线路径 $p(\pi_1, \phi_1) \to \sigma^{-1} \cdot p(\pi_0, \phi_0)$诱导的辫子，可利用引理 10 计算。输出 $\sigma_2\sigma_1\sigma_3$是由 F诱导的辫子，含义如定理 13 所述。