Office Hour Eighteen

Braids

Aaron Abrams

Clear your coil of kinkings
Into perfect plaiting,
Locking loops and linkings
Interpenetrating.

James Clerk Maxwell, from a poem sent to Peter Guthrie Tait, 1877

This office hour is about braids. No doubt you already have some idea of what a braid is; two familiar examples are shown in Figure 18.1.

Braids are among the oldest topics discussed in this book. Their history begins thousands of years ago,[1] and they entered the realm of mathematics at least several centuries ago. The modern mathematical study of braids involves lots of beautiful geometry and group theory, some of which I will introduce in this office hour. In addition to giving several different ways to think about mathematical braids and some of the basic theorems about them, I will also indicate some of the ways that braids relate to other parts of mathematics and science.

18.1 GETTING STARTED

Just as mathematical points, lines, and surfaces are idealized abstractions of physical objects, so mathematical braids are an idealized abstraction of the familiar hair

[1] Look up *quipu* to learn about the early history of braids.

UNIVERSITY LIBRARIES
UNIVERSITY OF NORTHERN COLORADO
GREELEY COLORADO 80639

Figure 18.1 Two braided things: Thalia, challah.

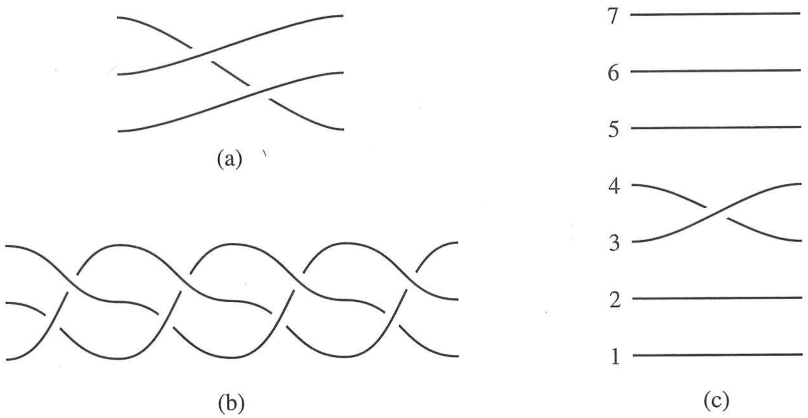


Figure 18.2 *(a)* A 3–string braid with two crossings. *(b)* The "usual" 3–string braid pattern. *(c)* String 3 crosses over string 4.

and bread braids of Figure 18.1. One important difference is what happens at the ends: in a mathematical braid, the "strings" remain separate at the ends rather than being fused together. The other main difference is that mathematical braids can have any number of braided strings and the braiding can occur in any pattern.

Here are some examples of mathematical braids:

When you braid hair or bread, usually you break it up into thirds, and then repeatedly pick up an outside piece and cross it over the middle piece. To make the usual braid pattern, you alternate picking up the two outside pieces, resulting in a braid whose mathematical form looks like Figure 18.2*(b)*.

Exercise 1. Draw the 4–string braid you get by taking the top string and crossing it over the middle two strings, and then taking the bottom string and crossing it over

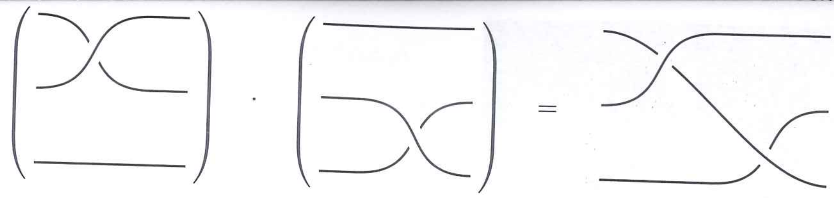


Figure 18.3 A product of two braids.

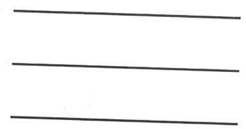


Figure 18.4 The identity in B_3.

Figure 18.5 Let's make these equivalent.

the middle two strings, and then repeating. What happens if you braid your hair this way?

From a mathematical point of view, there are a lot of interesting things about braids. One of the most interesting is the raison d'être for this office hour: for each fixed n, the set B_n of all n–string braids has a natural group structure.

The way you multiply two braids is by sticking them together, as in Figure 18.3.

What would it mean to have the inverse of a braid? For example, what's the inverse of the braid α shown in Figure 18.2*(b)*? Well, first you need to know what the identity is. For $n = 3$, the identity braid is shown in Figure 18.4.

So, how could you extend α to make it look like the identity? Literally, it's impossible: α has crossings in the picture, and drawing more stuff doesn't change the fact that you have crossings in the picture. What you need is a way to talk about canceling crossings, so that, for instance, the two braids in Figure 18.5 would be considered equivalent.

The idea is that you should be able to move the braids around as you can in physical space—so, without passing any string through another—and arrive at an equivalent braid. To define equivalence, we should also say that when you move the braids around, the endpoints have to stay fixed—otherwise any braid would be equivalent to the identity! It may help you to imagine that the left endpoints of each braid are fastened to the vertical wall $x = 0$ in 3–space, and the right endpoints are fastened to the wall $x = 1$. Now, two braids are *equivalent* if you can move one braid, keeping it between the walls and keeping the endpoints fixed, and

UNIVERSITY LIBRARIES
UNIVERSITY OF NORTHERN COLORADO
GREELEY COLORADO

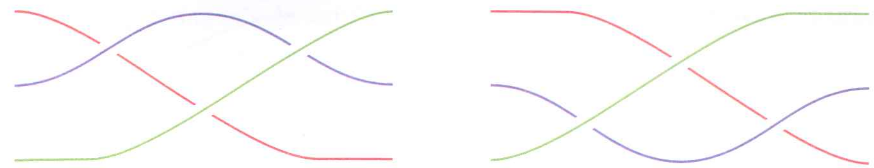


Figure 18.6 Two equivalent braids.

make it look like the other.[2] With this definition, the two braids in Figure 18.5 are equivalent. So are the two braids in Figure 18.6.

There is a technical point to make now. If we fix the bounding walls for the braid to be $x = 0$ and $x = 1$, then we should be a bit more careful about what it means to multiply braids. Precisely, to form β times β', we need to shift β' by one unit in the x–direction, take the union of β with the shifted version of β', and then scale the whole thing in the x–direction by a factor of $1/2$. This makes the product of two braids into a braid.

Exercise 2. *(a)* Convince yourself that the braid in Figure 18.4 is actually an identity element with respect to the product operation.
(b) Find a braid β such that α (the braid from Figure 18.2*(b)*) times β is equal (i.e., equivalent) to the identity.
(c) Describe a general procedure for constructing the inverse of a braid. *(Hint: Use a mirror.)*

It hasn't been said yet, but there is one more important feature of a braid: if you follow any individual string from the left endpoint to the right endpoint, you have to move monotonically from left to right. The strings can't ever "backtrack." So in particular no individual string can ever have a knot in it. Without this restriction, we would be able to make braids that don't have inverses!

There are no other rules. You now know the complete definition of a braid.

Exercise 3. Show that for fixed n, the (equivalence classes of) n–string braids form a group.

This group is called the n–string *braid group*. It's denoted by B_n and it's the primary subject of this office hour.

18.2 SOME GROUP THEORY

If you are trying to draw a braid, you can always use equivalence to avoid having three strings cross at the same point in the drawing. You can also make it so that no crossing occurs directly above another in the picture as in Figure 18.7.

Because of this, you can chop up any braid using vertical lines so that each chunk only contains one crossing, as in Figure 18.8.

[2] This is very similar to the notion of *isotopy* that you may have seen in knot theory.

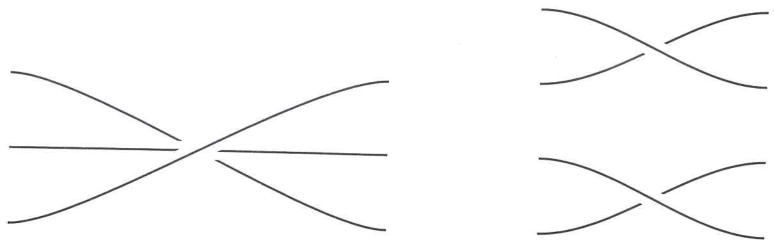


Figure 18.7 Drawings like these are avoidable.

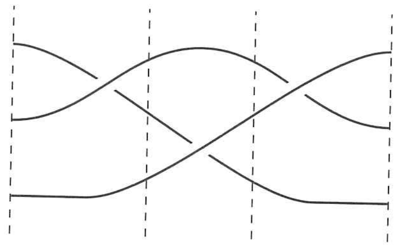


Figure 18.8 A braid as a product of crossings.

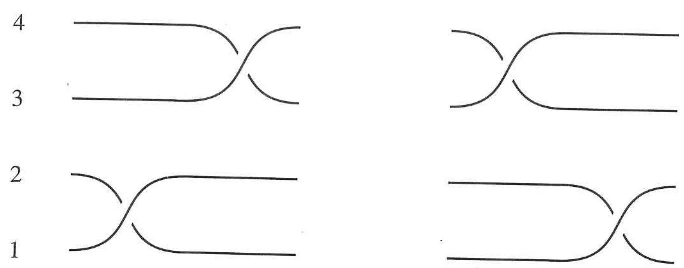


Figure 18.9 $\sigma_1\sigma_3$ is equivalent to $\sigma_3\sigma_1$.

In other words, B_n is generated by the set of all n–string braids that have exactly one crossing. The braid in which string i crosses over string $i+1$ is commonly called σ_i, so, for example, Figure 18.2*(c)* shows the braid σ_3 in B_7. Think about what its inverse is. You can now see that B_n is generated by the set $\{\sigma_1, \ldots, \sigma_{n-1}\}$; this is the braid group version of Exercise 1 from Office Hour 1. (Note that the notation σ_i doesn't tell you which braid group you're in, except that it must be at least B_{i+1}. The context usually clarifies this.)

Exercise 4. Write down each of the braids we've seen before as products of the σ_i and their inverses.

Let's think about how the different σ_i relate to each other. Notice first that if σ_i and σ_j involve completely different strings, then they commute; see Figure 18.9.

Exercise 5. Show that neighboring σ_i's do not commute: $\sigma_i\sigma_{i+1} \neq \sigma_{i+1}\sigma_i$.

UNIVERSITY LIBRARIES
UNIVERSITY OF NORTHERN COLORADO
GREELEY COLORADO 80639

The neighboring σ_i's do satisfy a relation, however. This is a famous equation called the *braid relation*:

$$\sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}.$$

(A similar braid relation in the mapping class group is discussed in Office Hour 17.)

Exercise 6. Draw a picture to convince yourself that the braid relation holds. Have you seen a picture like this before?

It turns out that these two types of relations, the commuting relations and the braid relations, imply all other relations that hold among the σ_i. In other words:

Theorem 18.1. *The braid group has the presentation*

$$B_n = \langle \sigma_1, \ldots, \sigma_{n-1} \mid \sigma_i\sigma_j = \sigma_j\sigma_i \text{ for } |i-j| > 1, \\ \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1} \text{ for } 1 \le i \le n-2\rangle.$$

We'll discuss how this theorem is proved later in this office hour.

Exercise 7. *(a)* Using Theorem 18.1, show that $B_2 \cong \mathbb{Z}$. Does this agree with your intuition?
(b) Using Theorem 18.1, write down a presentation for B_3.
(c) Show that $B_3 \cong \langle\, x, y \mid x^3 = y^2 \,\rangle$. *(Hint: Call the latter group G, and define a map from G to B_3 by sending x to $\sigma_1\sigma_2$ and y to $\sigma_1\sigma_2\sigma_1$. Show that this is well defined and an isomorphism.)*

Exercise 8. *(a)* Using Theorem 18.1, show that the abelianization of B_n is isomorphic to $\mathbb{Z}$ when $n \geq 2$.
(b) Describe the abelianization map from B_n to $\mathbb{Z}$. For example what number is the image of $\sigma_1\sigma_2\sigma_1^{-1}$?
(c) Define a function ℓ, called the *length homomorphism*, from B_n to the integers as follows. If w is a word in the generators σ_i and their inverses, set $\ell(w)$ to be the sum of the exponents of the σ_i in w. In other words, each σ_i counts for 1 and each σ_i^{-1} counts for -1. Show that ℓ is well-defined and a homomorphism.
(d) Show that ℓ is exactly the same function as the abelianization map.

Exercise 9. *(a)* Show that the generators σ_i of B_n are all conjugate to each other.
(b) Let $\delta = \sigma_1\sigma_2\sigma_1$ and let $\gamma = \sigma_2\sigma_1^{-1}$. Show that $\delta\gamma\delta^{-1} = \gamma^{-1}$. Thus the braid group contains an element γ that is conjugate to its inverse.
(c) Show that if γ is any braid that is conjugate to its inverse, then $\ell(\gamma) = 0$.

Permutations. Look back at the picture of a permutation in Section 1.1 in Office Hour 1. It looks a lot like the pictures of braids we are drawing, doesn't it? The only difference is that with a braid, you keep track of over and under crossings, whereas with a permutation there's just a crossing.

What this means is that there is a map $\pi : B_n \to S_n$ where you take a braid, ignore the overs and unders, and just read the corresponding permutation. (Recall that S_n stands for the *symmetric group*.)

Exercise 10. Show that $\pi : B_n \to S_n$ is a homomorphism.

By the way, how did you solve Exercise 5? Do you have any tools to show that two braids (such as $\sigma_1\sigma_2$ and $\sigma_2\sigma_1$) are different? Well, now you do: look at their images under the map π!

Let's use the symbol τ_i for the image $\pi(\sigma_i)$. When we ignore the crossings of a braid and turn it into a permutation, we are effectively saying that the image of σ_i is the same as the image of σ_i^{-1}. That is, $\tau_i = \tau_i^{-1}$, or equivalently $\tau_i^2 = 1$.

Exercise 11. Consider the following presentation, which resembles the presentation in Theorem 18.1:

$$G = \langle \tau_1, \dots, \tau_{n-1} \mid \tau_i\tau_j = \tau_j\tau_i \text{ for } |i-j| > 1,$$
$$\tau_i\tau_{i+1}\tau_i = \tau_{i+1}\tau_i\tau_{i+1} \text{ for } 1 \le i \le n-2,$$
$$\tau_i^2 = 1 \text{ for } 1 \le i \le n-1\rangle.$$

(a) Find elements $\tau_i \in S_n$ that satisfy all these relations. *(Hint: Use the preceding discussion.)*
(b) Use part *(a)* to show that the map taking $\tau_i \in G$ to $\tau_i \in S_n$ extends to a well-defined surjective homomorphism from G to S_n.
(c) Show that this homomorphism is also injective. Thus the above is a presentation of S_n.

Exercise 12. Write down the permutations associated to all the braids we've drawn so far.

In a way, this says that you can think of a braid as being like a permutation, except that when you describe it in terms of successive transpositions $(i\ i+1)$, you have to specify not just what i is but also *how* you are swapping i with $i+1$: does i go over $i+1$ or does it go under?

Pure braids. The kernel of the map $\pi : B_n \to S_n$ is an important subgroup of B_n. It is called the *pure braid group* PB_n, and it consists of all braids in B_n such that the strings line up in the same order on the left as they do on the right. (Being in the kernel of π means that they map to the identity permutation.) We will see a nice interpretation of PB_n in a little while.

You could also explore different subgroups of B_n that are related to π, but that aren't just the kernel. For instance, you could define an "even" braid to be an element of $\pi^{-1}(A_n)$, where A_n is the alternating group (consisting of the even permutations).

Exercise 13. Find an easy way to tell whether a braid is even.

Exercise 14. *(a)* What is the index of PB_n inside B_n?
(b) Recall that $B_2 \cong \mathbb{Z}$. Describe the subgroup PB_2.

Coming up with generators for the pure braid group is a bit trickier than for the full braid group, because you can't always chop up a pure braid into obvious chunks

UNIVERSITY LIBRARIES
UNIVERSITY OF NORTHERN COLORADO
GREELEY COLORADO

that are pure. Can you come up with a candidate generating set? For example, does the set $\{\sigma_i^2\}$ generate PB_n?

In fact there is a "standard" generating set for the pure braid groups that was defined first by Artin (see [11, 12]). There are $\binom{n}{2}$ generators called A_{ij}; there is one for each i and j with $1 \le i < j \le n$. These generators are defined in terms of the standard generators for B_n by

$$A_{ij} = (\sigma_{j-1}\sigma_{j-2}\cdots\sigma_{i+1})\sigma_i^2(\sigma_{j-1}\sigma_{j-2}\cdots\sigma_{i+1})^{-1}.$$

Exercise 15. Draw A_{ij} and describe it in words.

Can you see any relations between the A_{ij}'s? Some of them obviously commute, right? In fact Artin gave a complete presentation of PB_n using these generators, but his relations are a bit complicated. It turns out that these generators can be used in a presentation that has only *commuting relations*, i.e., relations that say that two elements commute. Mind you, these elements are not always just generators.[3] For example, the element A_{23} commutes with the element $A_{12}A_{13}$.

Exercise 16. Verify the last claim.

Project 1 at the end of this office hour invites you to study these and other presentations of the pure braid groups.

Combing and the word problem. Artin used the generators A_{ij} to describe a process he called *combing* for a pure braid. The idea behind combing is to take a pure braid and put it into a standard form, so that if you start with two pure braids that are equivalent (even if they don't look equivalent!) then you will end up with the same picture after combing. The normal forms we find here are in the same spirit as the normal forms found in Office Hours 14 and 16 for right-angled Artin groups and Thompson's group.

The combing process is iterative: roughly speaking, starting with a braid β in PB_n, you first remove the nth string to produce a braid called β_{n-1}. Then remove string $n-1$ to get the braid β_{n-2}. Eventually you get down to just a single string. Draw this as a straight line. Now reinstate the second string, creating β_2, but don't change how you've drawn the first string. (So the first string stays straight.) Then add in the third string, without changing the first two, being careful to *put any crossings involving the new string to the left of all previous crossings*. Continue like this. What you end up with is a picture of β that naturally decomposes into $n-1$ "blocks," where each block contains only one string that isn't straight. An example of a combed braid is shown in Figure 18.10, in which the vertical dotted lines delimit the blocks.

For a more detailed and truly excellent demonstration of braid combing, watch the animated video [99] produced by Ester Dalvit as part of her Ph.D. thesis. It is available at http://matematita.science.unitn.it/braids/index.html or on YouTube.[4]

[3] In particular, if $n \ge 4$ then PB_n is *not* a right-angled Artin group, although this is not so easy to prove. The subgroup of B_n generated by the elements σ_i^2, on the other hand, *is* a right-angled Artin group! See Office Hour 14 for more details.

[4] I wonder which will live longer: this book or YouTube?

UNIVERSITY LIBRARIES
UNIVERSITY OF NORTHERN COLORADO
GREELEY, COLORADO 80639

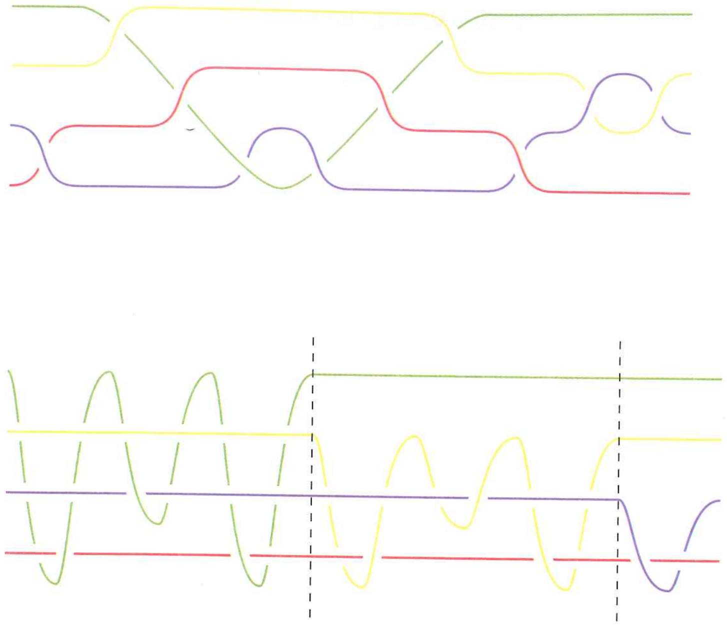


Figure 18.10 The result of combing. It's tricky; try it!

The part on combing starts about five minutes into Chapter 2, and lasts just a few minutes. In fact, I stole Figure 18.10 from her video: this is exactly the example she works out!

There are some things to point out about the combed braid. For one, the result is obviously not in its minimal form. However, this form makes it easy to write the braid in terms of the A_{ij}. Also, note that even after you remove the obvious cancellations, there are still a lot more crossings in the combed version than in the original! The fact that the combing is nevertheless useful illustrates the tenet that, depending on your goals, the "simplest" form of an object is not always the most convenient form to use.

Exercise 17. *(a)* Write the top braid in Figure 18.10 in terms of the generators σ_i.
(b) Write the bottom braid in Figure 18.10 in terms of the generators A_{ij}.
(c) Show that the two braids are equivalent.

The idea of combing helps us solve the word problem in B_n (refer to Office Hour 8 for a discussion of the word problem). Why? Well, Artin proved the following theorem about combed braids. See Artin's paper [11] for a proof.

Theorem 18.2. *If two pure braids are equivalent, then they have the same combing.*

In other words, the combed version of a braid is unique; two braids might not initially look the same, but if they are, then once they're combed they'll be identical. Because of this the combed version is called a "normal form" for the braid.

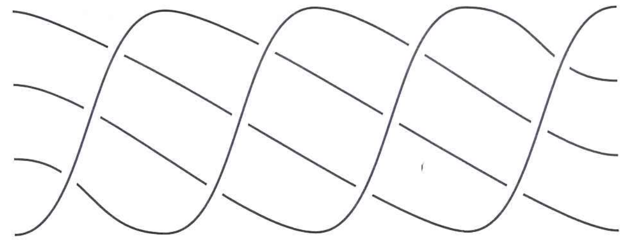


Figure 18.11 The full twist, Δ^2.

Exercise 18. Comb the braids $A_{23} \cdot (A_{12}A_{13})$ and $(A_{12}A_{13}) \cdot A_{23}$.

So what does this have to do with the word problem? Well, the problem after all is to determine for any given braid whether or not it is equivalent to the identity. So, given a braid, step one is to check whether or not it is pure. If not, it's not the identity! Thus we can suppose it's pure. Now comb it. By Artin's theorem, if it's equivalent to the identity, then the combed version will look exactly like the combed version of the identity, namely, the identity. So that's a solution to the word problem: as soon as we comb the (pure) braid, we know whether or not it's equivalent to the identity.

Theorem 18.3. *The braid groups have solvable word problem.*

This is just one of many known ways to solve the word problem in the braid groups, and in fact it's a relatively inefficient one: it takes exponential time as a function of the number of crossings in the braid. (Note, in particular, that the 10–crossing braid in Figure 18.10 has a combed form with 28 crossings! This gives a hint, though not a proof, that the combing process might not be particularly efficient.) There are now quadratic-time algorithms for this problem, meaning that the number of steps required to come up with the answer is a quadratic function of the length of the braid word. One of the projects at the end of the office hour is to learn about the many different solutions to the word problem for braid groups.

The twist. There is a braid called a *full twist* that you can imagine as follows: take the identity braid and spin the entire right wall by 360 degrees. When the endpoints return to their original positions, the braid created in the strings is the full twist. (Remember, you can't show the full twist is equivalent to the identity by unspinning the wall, because braid equivalence requires the endpoints to stay still throughout the process.) The full twist is denoted by Δ^2. See Figure 18.11.

Exercise 19. *(a)* Write down the full twist Δ^2 in B_n as a word in the braid generators $\sigma_i^{\pm 1}$.
(b) Show that Δ^2 commutes with every other braid in B_n.
(c) Show that, as the notation suggests, Δ^2 is indeed the square of a braid in B_n. (Note: Your argument should work regardless of whether n is even or odd.)

(d) The braid you found in part *(c)* is called the *half twist*, denoted by Δ. Does Δ also commute with everything in B_n?
(e) Show that Δ^2 has both a square root and an nth root.

Another way to say the conclusion of part *(b)* of Exercise 19 is that the full twist is in the *center* of B_n. In fact, Artin proved the following theorem in [11].

THEOREM 18.4. *If $n \geq 3$, then the center of B_n is an infinite cyclic subgroup generated by the full twist.*

In other words, any braid that commutes with all other braids must be a power of the full twist. (Why is this not true for $n = 2$?)

Exercise 20. Without using the fact that the center is generated by the full twist, show that the center of B_n is contained in PB_n.

Exercise 21. Prove Theorem 18.4 in the case of $n = 3$: the center of B_3 is generated by the full twist. *Hint: Use Exercise 7*(c).

Understanding the center is helpful for various reasons, foremost among them that the center (of any group) is a *characteristic* subgroup. This means that every automorphism (i.e., isomorphism) from a group G to itself takes the center to the center. In the case of braid groups we can use this, for example, to prove the following theorem, also due to Artin.

THEOREM 18.5. *If $B_m \cong B_n$, then $m = n$.*

This is a reassuring statement, isn't it? To prove this theorem, Artin used the abelianization map, or equivalently the length homomorphism ℓ as defined in Exercise 8. Remind yourself now what those are.

Exercise 22. Show that $\ell(\Delta^2) = n(n-1)$ if Δ^2 is the full twist in B_n.

Here is Artin's proof of Theorem 18.5. First note that we may assume $m, n \geq 3$, since $B_2 \cong \mathbb{Z}$ and B_n is non-abelian if $n \geq 3$, so B_2 cannot be isomorphic to B_n for any $n \geq 3$. Now, suppose $B_m \cong B_n$ with $m, n \geq 3$, and suppose $\phi : B_m \to B_n$ is an isomorphism. To minimize confusion, let's use Δ_i^2 to refer to the full twist in the group B_i. The center of B_i is infinite cyclic and generated by Δ_i^2, so because the center is characteristic, the isomorphism $\phi : B_m \to B_n$ must take Δ_m^2 to $\Delta_n^{\pm 2}$. Now, the map ϕ induces a map ϕ' on the abelianizations, and ϕ' is an isomorphism too. (See Exercise 23 below.) The map ϕ' takes the image of Δ_m^2 (in the abelianization of B_m) to the image of Δ_n^2 (in the abelianization of B_n). As ϕ' goes from $\mathbb{Z}$ to $\mathbb{Z}$, it must be multiplication by ± 1. Thus, by Exercise 22, we see that $m(m-1) = \pm n(n-1)$. Since m and n are positive, this implies $m = n$.

$$\begin{tikzcd}
\Delta_m^2 \arrow[r, mapsto, "\phi"] \arrow[d, "q"] & \Delta_n^{\pm 2} \arrow[d, "q"] \\
m(m-1) \arrow[r, mapsto, "\phi'"] & \pm n(n-1)
\end{tikzcd}$$

Exercise 23. If $\phi : A \to B$ is an isomorphism of groups, show that ϕ induces an isomorphism ϕ' from the abelianization of A to the abelianization of B.

UNIVERSITY LIBRARIES
UNIVERSITY OF NORTHERN COLORADO
GREELEY, COLORADO 80634

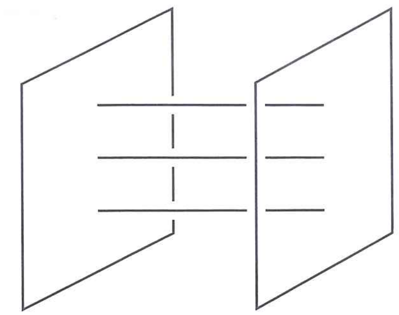


Figure 18.12 The identity braid is a boring movie: the particles sit still as the "time" coordinate x goes from 0 to 1.

18.3 SOME TOPOLOGY: CONFIGURATION SPACES

There are numerous important connections between braid groups and topology, and in this section we discuss one of the most basic ones. The central concept is that of a *configuration space.*

The phrase "configuration space" generally refers to a space that encapsulates all the possible states of a complex system. Configuration spaces are also sometimes called state spaces or parameter spaces. For example, configuration spaces can be used to model the collective motions of several objects, such as cars on city streets or packets in a network or molecules in solution or robot workers in a factory. For a perhaps more concrete example, consider the three joints in your arm: the shoulder, elbow, and wrist. Your shoulder and wrist each have two degrees of rotational freedom, and your elbow has one, for a total of five dimensions of configurations. A robotic arm modeled on yours has to navigate through a five-dimensional configuration space in order to take advantage of all the flexibility built into this joint structure. To illustrate this, let's imagine you don't have a hand, but rather just a rigid platform attached to your wrist. Suppose you want to lift a full glass of water from below your waist to above your shoulder. Can you simply rotate your shoulder joint? This is what babies do, and they spill the water every time! Eventually, they learn how to coordinate the shoulder rotation with the requisite elbow and wrist motions to keep the glass upright. Once you're good at it it seems easy, but the five-dimensional configuration space takes some getting used to.

So, what is the connection with braids? We have been talking about individual braids as topological objects, but it is also true that the braid *groups* are topological objects, in the sense that each braid group describes, in a natural way, the different types of loops that exist in a certain configuration space. The space in question models the collective motion of n distinct particles in the plane that are not allowed to collide.

Let's get specific. Begin with n distinct particles in the plane. Stand the plane up so that it forms the left wall of a braid. Now slide the plane from left to right, and imagine that each particle leaves a trail behind it in space. What happens?

Thus β^2 is indeed right veering. By repeating this argument we can now easily show that β^n is right veering for all $n > 0$, as desired. □

Thus the braid groups B_n are torsion free. One lesson to learn from this set of ideas, and from this book in general, is that a topological or geometric approach can often provide a lot of insight into algebraic results. To be fair, there are some details missing from this proof. For example, homeomorphisms can be complicated: for one thing, the arc c_0 might intersect a_0 infinitely many times. Then how do you reduce it? This and other details can be dealt with in a completely rigorous way, but once people get the hang of these types of technicalities, they tend to find it more palatable to omit them than to include them.

The word problem revisited. We are now primed to see several more approaches to the word problem. The first is an exercise.

Exercise 39. Using Theorem 18.8, describe a solution to the word problem in B_n using curve diagrams.

To delve further, see Projects 2, 4, and 5 at the end of the office hour.

18.5 CONNECTION: KNOT THEORY

Having proved some basic theorems about braid groups, we will now discuss some further connections between braids and other areas of mathematics. The first of these is knot theory.

If you have ever studied knot theory, then you have probably already noticed some similarities between knots—or more generally links—and braids (a link is just like a knot, except that the number of components is allowed to be greater than 1). For instance, they are both defined in terms of pictures of strings in 3–space, and they both have a notion of *equivalence* (usually called *isotopy*) that describes when pictures are equivalent. Of course there are also some differences, most notably that braids have endpoints that aren't allowed to move during isotopies. Another difference is that each string of a braid goes monotonically from left to right, whereas there is no such restriction with links.

In fact, one of the early motivations for studying braids was to help understand the theory of (knots and) links.[6] The connection is that every braid can be "closed up" to form a link. It was hoped that the group structure on braids would lead to some sort of group structure on the set of links. It turns out that that doesn't work, but there are nevertheless other ways to use braids to help with the study of links.

Here is how you close up a braid: just connect the left endpoints to the right endpoints without introducing any new crossings. Suddenly you have a link!

[6] The quote that opens this office hour is part of the poem, *(Cats) Cradle Song*, written by Maxwell in response to Tait's manuscript *On Knots*, the first mathematical treatise on knot theory. According to Silver [241], the word "coil" refers to what we now call a braid.

UNIVERSITY LIBRARIES
UNIVERSITY OF NORTHERN COLORADO
GREELEY, COLORADO

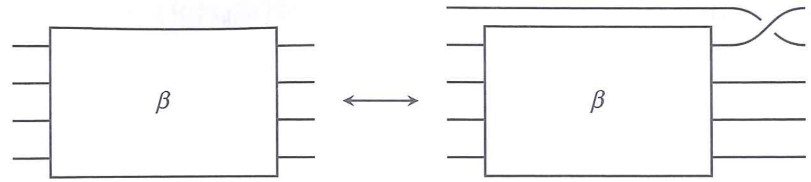


Figure 18.18 The Markov move: here β stands for an arbitrary braid (with any number of strings). The Markov move adds (or subtracts) a string and a single crossing.

Exercise 40. *(a)* Show that when you close up the 3–string braid $(\sigma_1\sigma_2^{-1})^2$, you get the Figure 8 knot.
(b) Show that the Borromean rings can be realized as a closed braid (look this up if you don't know what it is).
(c) Show that every link can be realized as a closed braid.

Part *(c)* of the preceding exercise is sometimes called Alexander's theorem. If you are stuck, Chapter 3 of the movie [100] illustrates one way to prove it.

One good thing about this theorem is that it gives you a straightforward way to describe any link over the phone (or any time you can't draw a picture). Rather than saying, "It's knot number 10_{161} in the standard table," which only helps if you have a standard table and you can find your knot in the table and you are talking to someone who has the same table, or "It goes over and then under, and then back around to where it was before only on the other side, and then . . . ," which is the sort of thing people say and which has some rather obvious drawbacks, you can instead simply say, "It's the knot you get by closing up the 3–string braid $\sigma_1\sigma_2^{-1}\sigma_1\sigma_2^2\sigma_1^2\sigma_2^3$." Much better, right?

Alexander's theorem doesn't give a one-to-one correspondence between links and braids, though. Lots of different braids can close up to give the same link, which is obvious once you think about it for a moment. (Think about it for a moment.) You can even have braids with different numbers of strings that close up to give the same link.

Exercise 41. How many different 2–string braids close up to the unknot? What about 3–string braids?

Exercise 42. Let β and γ be n–string braids. Show that when you close up β, you get the same link as when you close up $\gamma\beta\gamma^{-1}$.

Conjugating a braid, as in the previous exercise, doesn't change the number of strings. Here is a move that does: it is called a Markov move. See Figure 18.18.

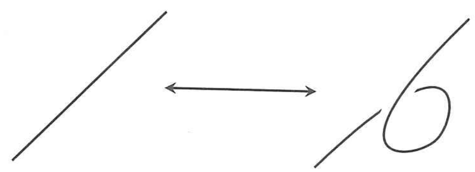
Exercise 43. Show that these two braids close up to give the same link.

In fact it is a theorem that any time you have two braids that close up to the same link, you can get from one braid to the other by a sequence of conjugations and Markov moves. You don't need any more operations besides these two. (This theorem is attributed to Markov although he may not have actually proven it.)

In the modern era, the connection between braids and links led to the discovery of the Jones polynomial, which is one of the most successful tools for distinguishing knots and links. To learn more about knots, braids, and knot polynomials including the Jones polynomial, see [4] or [182].

Aside: Reidemeister moves. If you have studied knots before, you must know about Reidemeister moves. If not, you could either skip the rest of this section or else go look up Reidemeister moves on the internet (for example in Wikipedia). In summary, these are three different types of changes you can make to a knot diagram that don't change the actual knot. Moreover they have the wonderful property that you can get from any diagram of a knot to any other diagram of the same knot by a sequence of Reidemeister moves. Only recently, a bound has been proven on how many moves this could take (as a function of the number of crossings in the diagram). This implies a conceptually simple algorithm for determining whether two diagrams represent the same knot: try all possible sequences of moves on one of the diagrams, and see if you ever produce the other diagram. Once you try all possible sequences with no more moves than the known bound, you know that if you haven't seen the other diagram, you never will. Unfortunately this isn't yet a practical algorithm, as the bound is still tremendously large.[7] Nevertheless, Reidemeister moves are also useful for a lot of theoretical purposes, like establishing many important knot invariants.

For braids, there is an analog of Reidemeister moves; namely, the Reidemeister moves themselves! Except that you don't need the Type I move, because of the condition that each string is monotonic. This means you can never have a loop, so you can never do a move of Type I.



What about Type II? If you see a part of a braid that looks as in Figure 18.5, then you can straighten both strings. This is a Type II move. Algebraically, it corresponds to cancelling $\sigma_i\overline{\sigma}_i$ (or $\overline{\sigma}_i\sigma_i$) out of a word, which of course you can do. You can also insert such a cancelling pair any time you want to without changing the braid.

Finally, what about Type III? Well, look closely at the picture you drew for Exercise 6 (or Figure 18.6). It's a Type III Reidemeister move on braids!

The point is that using Type II and III Reidemeister moves, you can get from any braid diagram to any other diagram of an equivalent braid. (Notice that these moves don't change the *parity* of the number of crossings in the braid. This is related to Exercise 13.)

[7] For details see the articles [92, 161, 190].

UNIVERSITY LIBRARIES
UNIVERSITY OF NORTHERN COLORADO
GREELEY, COLORADO

18.6 CONNECTION: ROBOTICS

Let's turn to a connection that has concrete real-world applications. The subject is again configuration spaces, but this time we will study particles moving around on a graph. These spaces are useful models for network traffic and various automated processes involving many moving parts. We will refer to the particles as *robots*. If you are in a context where it is important that the robots (or cars, or packets, or whatever) not crash into each other, then what you really want to study is the configuration space. To safely coordinate the motions of several robots, you really ought to understand the geometry, i.e., the paths and loops and so on, in a configuration space.

Let G be a finite graph, made of vertices and edges. We have already defined the configuration spaces, both ordered and unordered, of n robots on G:

$$C_n(G) = \{(p_1, \dots, p_n) \in G \times \cdots \times G \mid p_i \neq p_j \text{ if } i \neq j\},$$

and $UC_n(G)$ is the unordered version of n-point subsets of G^n. There are, correspondingly, pure graph braid groups $PB_n(G)$ and graph braid groups $B_n(G)$ defined as the fundamental groups of these spaces, or, in other words, the equivalence classes of based loops of (labeled or unlabeled) configurations.

What does a nontrivial element of $PB_n(G)$ look like? Such an element is a loop in the configuration space $C_n(G)$, but more concretely it is a "nontrivial" motion of n robots on G. How do you picture what nontrivial means?

Exercise 44. Let Y be the graph with one vertex O connected by edges to each of the three vertices A, B, and C, and with no other edges or vertices. Act out a nontrivial braid in $PB_2(Y)$.

Discretization. Visualizing these configuration spaces can be tricky, even when $n = 2$. But there is a technique called *discretization* that helps a lot. It works like this: instead of allowing the robots to move anywhere they want on the graph (subject to the rule that they don't collide), we will restrict their motions somewhat. Imagine the vertices of the graph as "stations." Each station has a manager who is responsible for any robot that is either at the station or on (the interior of) an edge that meets the station. Thus a robot at a station occupies just one station manager, but a robot between two stations (meaning, on an edge) occupies the managers at both end stations of the edge.

Now consider all (ordered) configurations of robots where no station manager is ever occupied by more than one robot. This is a subset of $C_n(G)$, and it is called the *discretized configuration space* $D_n(G)$. Similarly, if you start with the unlabeled configurations $UC_n(G)$ and follow the same procedure, you get the *unordered discretized configuration space* $UD_n(G)$.

Let's consider an example.

Example 18.6.1. *Let G be the graph Y from Exercise 44, and consider $D_2(Y)$. Then $D_2(Y)$ has 12 vertices, corresponding to the possible locations of the two robots. From the vertex AB, there are edges leading to OB (corresponding to moving*

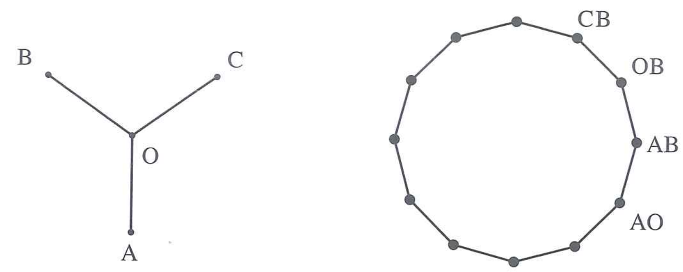


Figure 18.19 The space $D_2(Y)$ is a 12–gon. Exercise: finish labeling the vertices.

the first robot) and AO (corresponding to moving the second robot). Continuing in this manner, we construct the space shown in Figure 18.19.

Exercise 45. *(a)* Fill in the details of this example.
(b) Describe the loop in $D_2(Y)$ that you followed when you did Exercise 44.

You can now see why this is called discretizing: the motions of the particles are being modeled as discrete motions. There are several other equivalent ways to define $D_n(G)$ which further explain the term. These are based on the idea that the graph G is a one-dimensional cell complex (with vertices and edges), and so the product G^n is an n–dimensional cell complex. In fact, each (closed) cell of G^n is of the form $c_1 \times \cdots \times c_n$, where c_i is either a vertex or an edge of G. In particular, each cell of G^n has the combinatorial structure of a cube (of some dimension $\leq n$). (Think of the edges as being "closed," meaning they include their endpoints.)

Exercise 46. *(a)* Show that $D_n(G)$ is the largest subcomplex of G^n (that is, the largest union of closed cubes in G^n) that is contained in $C_n(G)$.
(b) Show that a cell $c_1 \times \cdots \times c_n$ of G^n is in $D_n(G)$ if and only if the (closed) cells c_i are pairwise disjoint in G.
(c) Show that an i–dimensional cell of $D_n(G)$ is a product of i (closed) edges and $n - i$ vertices of G that are pairwise disjoint.
(d) Show that any loop in $D_n(G)$ can be moved continuously until it only uses the edges of $D_n(G)$. This means that, although it is legal for several robots to be moving at the same time, it is always possible to exchange such a motion for an equivalent one where only one robot is moving at a time.

You can now see why the space $D_2(Y)$ is one-dimensional: the graph doesn't contain any pairs of disjoint edges, which is what you would need to get two-dimensional cells in D_2.

Exercise 47. *(a)* In terms of the graph G, how many vertices does $D_2(G)$ have? How many edges?
(b) Draw $D_2(G)$ for various graphs G: the letter X, a triangle, a square, a pentagon, the complete graphs K_4 and K_5, and any others you'd like to try. In each case, can you predict how many squares $D_2(G)$ will have?

Drawing $D_2(K_5)$ is a bit tricky: it can't be drawn in the plane, but it can be drawn in 3–space. It is hard to resist including a picture, but you should do it yourself!

UNIVERSITY LIBRARIES
UNIVERSITY OF NORTHERN COLORADO
GREELEY COLORADO

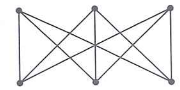


Figure 18.20 The utilities graph. Look up the "utility problem" to learn the origin of the name.

Exercise 48. If you gave up on drawing $D_2(K_5)$, try again. What does a neighborhood of a point look like? *Hint: The space is a closed surface. Which surface is it? (Use the Euler characteristic, introduced in Office Hour 17.)*

Exercise 49. The "utilities graph" $K_{3,3}$ is shown in Figure 18.20. Draw $D_2(K_{3,3})$. *Hint: This one is also a closed surface! Which surface is it? (Use the Euler characteristic again.)*

No graph other than K_5 and $K_{3,3}$ yields a closed surface when you construct D_2. Not only that, but for $n > 2$, it is never the case that $D_n(G)$ is a closed n–manifold. So these two examples are really quite special! For more on these and various other examples, see [3] or [2].

At the beginning of this section we mentioned that the process of discretization provides a helpful way to visualize the configuration spaces $C_n(G)$. You may have noticed that we never really justified this: what is the relationship supposed to be, other than that D_n is a subset of C_n? The answer is provided by the following theorem.

Theorem 18.11. *If the graph G is subdivided enough, then $D_n(G)$ is a deformation retract of $C_n(G)$.*

This means that $C_n(G)$ is basically a "thickened up" version of $D_n(G)$. In particular, they have the same fundamental group, so our implicit identification of the graph braid group $B_n(G)$ with the fundamental group of $D_n(G)$ (rather than $C_n(G)$) is justified, provided that the hypothesis of the theorem holds. What does "subdivided enough" mean? Insert enough valence 2 vertices along each edge so that any motion you can imagine in $C_n(G)$ can be approximated by a motion in $D_n(G)$. This is subdivided enough. In particular, if there are two robots, it is enough to have a "simple graph," which is a graph that has no loops consisting of just one or two edges. In general, what you actually need is enough stations so that all the robots can fit on any given edge at the same time, plus one extra on each loop so that there is room to move around.

18.7 CONNECTION: HYPERPLANE ARRANGEMENTS

We will now describe one more geometric connection with braid groups. Do you know what a hyperplane in a vector space is? It is a linear subspace of codimension 1, i.e., of dimension 1 less than the dimension of the vector space. In a real vector space $\mathbb{R}^d$, every hyperplane is isomorphic to $\mathbb{R}^{d-1}$. If you remove a hyperplane from $\mathbb{R}^d$, you are left with two connected components, each of which is an open half-space. What happens if you remove more than one hyperplane?

Exercise 50. *(a)* If you remove k hyperplanes from $\mathbb{R}^2$, how many components are left?
(b) If you remove two hyperplanes from $\mathbb{R}^3$, how many components are left?
(c) If you remove three hyperplanes from $\mathbb{R}^3$, how many components are left?
(d) If you remove two hyperplanes from $\mathbb{R}^4$, how many components are left?

Actually, this exercise is a bit of a trick question. The number of components left over when you remove k hyperplanes from $\mathbb{R}^d$ doesn't just depend on k and d. It also depends on how the hyperplanes intersect each other. One part of Exercise 50 has multiple possible answers. Do you know which part?

Anyway, again we ask the question: what does this have to do with braids? The idea is to think about complex vector spaces instead of real ones. The linear algebra doesn't change: a hyperplane H in $\mathbb{C}^d$ is isomorphic as a vector space to $\mathbb{C}^{d-1}$, and it has dimension $d-1$ as a vector space over $\mathbb{C}$. But as a topological space, the dimension of $\mathbb{C}^d$ is $2d$, and the dimension of H is $2d-2$. It has "complex codimension 1" but "real codimension 2." What this means is that if you delete a hyperplane from $\mathbb{C}^d$, the result remains connected! It is like deleting a line from $\mathbb{R}^3$: instead of losing connectivity, the space loses simple-connectivity (as in Office Hour 8, a path-connected space is *simply connected* if its fundamental group is trivial, or if every loop is homotopic to a point). A hyperplane H in $\mathbb{C}^d$ has a "linking circle" that wraps around H exactly once; this circle generates the fundamental group of $\mathbb{C}^d - H$, which is isomorphic to $\mathbb{Z}$.

Exercise 51. *(a)* Convince yourself that if you start with the space $\mathbb{C}^2$, with complex coordinates z and w, and you delete the hyperplane $z = 0$, you are left with a space X that is connected.
(b) Convince yourself that the unit circle $|z| = 1$ in the hyperplane $w = 0$ is *(i)* contained in X and *(ii)* a *linking circle* for the deleted hyperplane $z = 0$. One way to see *(ii)* is to show that the circle doesn't bound any disk that is disjoint from the hyperplane $z = 0$.

That's the situation with one hyperplane. If you delete a bunch of hyperplanes from $\mathbb{C}^d$, you always get a connected space but the fundamental group can be quite complicated.

So now we see how this relates to braids. As we learned in Section 18.3, the pure braid group PB_n is the fundamental group of the configuration space $C_n(\mathbb{R}^2)$. If we identify $\mathbb{R}^2$ with the complex numbers $\mathbb{C}$, then we can write PB_n as the fundamental group of the space

$$\begin{aligned} C_n(\mathbb{C}) &= \{(z_1, \ldots, z_n) \in \mathbb{C}^n \mid z_i \neq z_j \text{ if } i \neq j\} \\ &\subset \mathbb{C}^n. \end{aligned}$$

This is the space of pairwise distinct n–tuples of complex numbers. It is a subset of the vector space $\mathbb{C}^n$. What pairwise distinct means is that z_1 is not allowed to be equal to z_2 or z_3 or any of the others, and so on. But look: the (linear) equation $z_1 = z_2$ describes a hyperplane in $\mathbb{C}^n$. So $C_n(\mathbb{C})$ is exactly what you get if you start with $\mathbb{C}^n$ and delete the $\binom{n}{2}$ hyperplanes $z_i = z_j$.

UNIVERSITY LIBRARIES
UNIVERSITY OF NORTHERN COLORADO
GREELEY COLORADO

Figure 18.21 A braid in the wild.

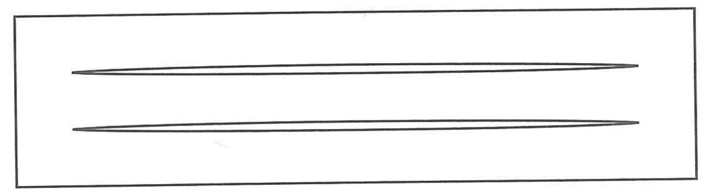


Figure 18.22 Slit your paper.

Thus the pure braid group PB_n can be viewed as the fundamental group of the complement of a collection of hyperplanes in $\mathbb{C}^n$. It turns out that lots of other interesting groups also arise as fundamental groups of complements of "hyperplane arrangements." To learn more about this subject, see [215] (or Wikipedia to get started).

18.8 A STYLISH AND PRACTICAL FINALE

Have you ever seen a braided belt, like the ones in Figure 18.21?

Did you ever wonder how they're made? Here's the basic idea. Take a strip of paper and cut two slits in it as in Figure 18.22.

Now, by braiding part of the strands and then passing one end through the slits, possibly several times, see if you can get a braid into the strip (without any further cutting). You want the individual strands to lie flat, i.e., face up, the whole time. In practice it is easier if the slits are long and thin and numerous. Try it with three, four, five, six, or more.

To carry out a similar experiment with your hair, the situation is slightly different, because there is no restriction that the strands "lie flat." Imagine braiding your hair by first making a ponytail and then doing the braiding. Which braids can you make this way? See Project 8.