

PROFESSOR: Let's get started. Today we're really going to take on folding polygons into polyhedra in a big way. We got started last time. Today we're going to do the real thing with some tape and scissors.

Folding polyhedra. It's pretty much the last kind of folding that we're going to be talking about in this class. We see linkage folding, paper folding, unfolding polyhedra, and finally, folding polyhedra.

So this is a, sort of, problem. We're given some polygon. We're not told where to folder or how to glue it up, necessarily. We'd like to find, is there some way to glue this boundary to itself-- I've drawn this picture before, something like this-- in order to make a convex polyhedron?

There's several versions of this problem. One is the decision version, which is, I give you a polygon. And I want to know, does it make any convex polyhedron? For the cross, the answer is yes. We will see examples later today where the answer is no. In fact, most of the time the answer is no.

So given a polygon, can its boundary be glued to itself? And this is the distinguishing feature from origami is that's all you get to do and no multiple coverage. You want exactly that surface to be folded into convex polyhedron.

And today's lecture is going to be mainly about mathematics for solving this problem. The next class will be algorithms for solving this problem. So we need some mathematical tools, some cool geometry stuff by a Russian geometer called Alexandrov, from the '40s. And that's what we'll be talking about today. Then we'll see how that turns into an algorithm next class.

Decision is nice. But of course, we'd like to actually find these gluings. I want to actually know that this guy folds into a cube.

So there's the enumeration problem, which is list all the gluings and all the foldings. And there's also the more mathematical combinatorial problem, which is, how many

can there be?

And we'll start addressing that today by showing that there can be infinitely many. There can be continua. For example, this polygon, this rectangle, can fold into infinitely many polyhedra. We'll prove that today. We won't do all of them. But we'll do one of them, I guess.

So that's makes this listing all of them a little bit difficult, when there's infinitely many. But it turns out they'll fall into some nice classes. We'll be able to characterize that at the end of today's class. And so this problem is actually well defined. They're a finite number of distinct classes. And you could list them all.

There can actually be exponentially many of those classes. But most the time, it's polynomial. That'll be more next class. These are the sorts of problems we're interested in folding.

Now, one natural question is, why am I restricting myself to convex? And one answer is, well, that's where most of the work has been done. But there is some nice work on the nonconvex case. So let me tell you about that.

So for nonconvex-- or where you don't care about whether the result is convex-- you just want it to make some polyhedron. You can show it's always possible.

What this means is if I'm given some polygon, and I glue up the boundary somehow-- and this result even holds if you don't glue up all of the boundary. Because then we're no longer trying to make a convex thing.

For example, I could glue up all the boundary, as long as the way that I glue things is orientable-- I don't put twists in an unresolvable way, in these gluings-- then you're golden. So you need that the surface is orientable or has boundary.

So this is posed as an open problem in our book. And then we discovered, this year, that it was already solved in 1960, but in Russian. And then, it was only translated in 1996. We still should have known about it in '96.

But we finally realized that they didn't talk about folding or gluing. They were talking about realizing polyhedral metrics, which is the language we're going to translate to soon. But our bad.

And I think this is a pretty complicated result. So we're not going to talk about it here. But it's interesting. I'd like to learn more about it. So it's by Burago and Zalgaller, I've used others of their theorems. So it's funny I didn't know this one.

In some sense, the decision question isn't interesting for non-convex polyhedra. So by restricting the convex, saying, I want to make convex shapes, this becomes an interesting problem. And it leads to lots of cool mathematics and algorithms. And that's what we're going to talk about.

So I want to talk about this guy, Alexandrov. Not really the guy, but the mathematics that he did. And we're going to appropriate his name a little bit and talk about Alexandrov gluings. This is a term that we made up, obviously he didn't. You don't usually make up your own terms named after yourself.

We'd like to know which gluings are valid. We're not going to think about where the thing is creased, how to fold. We're just going to focus on the gluing for now. What makes a gluing valid?

Well, one thing that we talked about last time is that once you have a gluing, in this picture, you can compute shortest paths. You can compute the shortest path from this vertex to this vertex. But here, it's a straight line. There's probably another path that goes around a different way.

Last time we did the shortest path from here to here. And it was some complicated thing. I won't try to do it again. But you can compute shortest paths once you know what's connected to what.

So this defines what we call a metric. We're going to learn something about that. Polygon plus gluing induce a metric. Metric just means it tells you how far different points are from each other. And we can compute that metric by computing shortest paths.

I'm going to start abstracting a little bit away from polygon plus gluing and just think about, there's some surface that, in your mind, it's not really geometric thing entirely. But locally, you understand the geometry. If you're at some corner, you can understand what other vertices are near to you and how to walk around on that surface.

So locally, you understand the metric and how it's laid out. When you're at this corner, you know-- and let's say I glued this to this-- that your curvature is 90 degrees, that there's 270 degrees of material there. And you could go any one of those 270 degrees of directions from there. So that's the sense in which we have a metric.

Now, there are three things that this metric should satisfy if we want to make something that's convex. Any suggestions?

AUDIENCE: Less than or equal to 360.

PROFESSOR: Less than or equal to 360 degrees of material an any point. Yep, that is what we would call convexity. It's really a local convexity. All points have 0 or positive curvature. That's the way I would say it.

Remember, curvature was 360 minus some of the angles at the vertex. So this is the same as saying you have, at most, 360 degrees of material any point. This is obviously necessary.

And the point is, you can evaluate this just knowing how things are glued together. You don't need to understand 3D geometry. It's a local property. And certainly necessary if we're going to try to make a convex polyhedron.

Anything else? They get less obvious from here, I guess. Any topologists?

AUDIENCE: You shouldn't have holes.

PROFESSOR: Shouldn't have holes. Right. So I should definitely glue everything together. It's sort of implicit in this statement but not totally explicit. And there's a little more than that.

Not only should everything be glued together, but for example, if I glued this edge to this edge and I glued this edge to this edge-- something like that. That's probably not going to work out. Crossings are bad.

And what's really going wrong there is the topology. If we're trying to make something that's convex, it better be topologically a sphere. So that's the third condition.

And this is, again, something you can figure out just by looking at the local metric. This is a property about metrics. In their case of polygons plus gluings, it's really easy to figure out when this is the case, which is that the gluing should be complete, as you were saying. You shouldn't leave anything blank.

Everything gets glued to something. And it has no crossings that will be equivalent to being sphere like, in our case.

The other property actually always holds. So it's no surprise that you didn't think of it. But if you want to think a little more generally, this is a key property we're using that we started from a polygon, not some smooth curve thing, which is that only a finite number of points have nonzero curvature.

If we want to make a polyhedron with a finite number of vertices, well, these are going to be the vertices, all the points with nonzero curvature. So there had better be a finite number of those. Now, when we start with a polygon and just glue some finite set of segments to each other, that will happen automatically. But it's a property we're using.

So these three properties together are what I'm going to call Alexandrov gluing. If you satisfy all three of these, you are Alexandrov. And now, we have a cool theorem about Alexandrov gluings called Alexandrov's Theorem. He had a lot of gluings, but this is the one we use a lot in this field.

Originally proved in 1941, I believe. Then finally, his big book about convex surfaces was translated to English in 2005, pretty recent. But we've been reading other

translations, and so on, to understand this theorem.

I guess I'm not using Alexandrov gluing here. But we have a convex, polyhedral, and topologically sphere. Take any one of those things. Then it's realized by a unique convex polyhedron.

So this is basically the answer. And from our perspective, it's like , yeah, yeah, polyhedral. Yeah, no crossings. But really, the central constraint that we have to worry about is convexity, that we never glue together too much material.

As long as you don't glue together more than 360 degrees material at any point, then that thing will realize a unique convex polyhedron. It's pretty cool but surprising. The uniqueness should not be surprising, because we proved that last class.

I can remind you briefly. We proved Cauchy's Rigidity Theorem, which is not quite the same. And then we said, oh, yeah, if you have some convex polyhedron metric, you don't know necessarily know what it looks like in 3D. But you can compute all shortest paths from every vertex to every other vertex.

And then however you try to fold that thing, every edge is a shortest path. That's just a fact about polyhedra. The edges must be some subset of these shortest paths. So you think of this as a convex polyhedron. And then Cauchy's Rigidity Theorem says, there's only one convex realization of that thing with those edges.

And Cauchy allows some of the edges to be flat like this. So there will only be one way to embed that in 3D-- at most one way. The surprising thing is that there actually is one way that you can realize-- I'm just thinking totally locally in this kind of gluing-- and yet, as long as I don't glue together too much material at any point, I get a convex polyhedron.

With the usual gluing here, without these crossings, I get a cube. No surprise, and it's unique. But there are other ways to glue this polygon. We saw them, actually, in Lecture 1-- I'll show them again next class-- that can make other convex polyhedra.

Let me give you a little idea of the proof. This is going to be a pretty high-level sketch. But I'll tell you the combinatorial part. And then there's the topology analysis part that's a little less obvious.

We've already proved uniqueness. I want to look at existence. And we're going to use induction on the number of vertices. And remember, vertices are well defined. When I say vertex here, I really mean point of nonzero curvature.

We know there's finitely many of them. Call the number n . And we're going to try to simplify our polyhedron and reduce the number vertices. But there's lots of ways to do that with polyhedra. But here, we don't have a polyhedron.

We have this kind of abstract polyhedron in Never Never Land. We know how to walk around on the surface, locally. But we don't really know any 3D geometry. So is this a crazy idea.

Before we get to the crazy idea, I need an important fact, which is total curvature, if you sum over all vertices, is 720 degrees. This is actually true, even of nonconvex polyhedra, as long as you have something homeomorphic to a sphere, topologically a sphere. Total curvature is 720.

This is actually why it's useful to talk about curvature and not how much angle is coming together at a point. So obviously, we've got zillions of points with a curvature 0. Those don't count for anything.

But then, somehow this thing has to wrap around a sphere. And you can compute the curvature of a sphere at 720. And this has to match that. I can't give a lot of intuition for this, but it's true. And it's kind of powerful.

It says you can't have very many sharp vertices. If I have a super-sharp point in my convex polyhedron, it has a curvature close to 360. It's going to be 260 minus almost 0. I can only have two of those in any polyhedron.

Now, with a nonconvex polyhedron, I can have negative curvature to compensate. And so then, I can have lots of spikes, like an our witch's hats, where we had four

spikes. But for convex, I don't have any negative numbers to compensate with these big positive numbers. So I can only have two spikes.

We're going to use that a lot today, this fact. It's called Descartes Theorem. You may have heard of Descartes. Nowadays, you prove it in other ways. But we're not going to look at a proof. We're just going to use it.

I'm also going to assume that my polyhedron has at least five vertices. Or really, my polyhedral metric has at least five vertices. If it has at least five vertices, in other words, more than four, then there will be some vertex that has less than a quarter of 720 curvature, in fact, two of them.

Two vertices-- and I call them x and y -- have curvature. Call them α and β , less than 180, which is one quarter of 720. The average is obviously going down as n goes up. So when n is 5, I can get two reasonably small curvature vertices.

This is like saying, there's more than 180 degrees of material glued at these two points So what?

Well, now I'm going to do some surgery, as topologists call it. I have these vertices, x and y , in this sphere-like thing. There's some shortest path between them. I can compute that. And now we're saying that there's a fair amount of material here and a fair amount of material here.

What I'd like to do is slice the surface open along that shortest path. I've drawn it twice, because I've cut it open. There's surface here, surface here, and a hole there. And I'm going to glue in some more material.

And I'm going to glue in a triangle-- actually, two triangles right on top of each other. So that here, the edges are glued together. Back here, the edges are glued together, here, they're not.

So you can see, this is still topologically a sphere. As you walk on to this edge, you end up on the top triangle. I can go around the back to the bottom triangle. And eventually, I will get to the back edge.

And then I can walk like that. Shortest paths are not preserved. But it's at least still topologically a sphere. I've slipped this open, wedged in two triangles. I get a new metric.

Now, I get to choose what triangles I glue in. And I'm going to set these angles. Ultimately, I'd like this angle to be $\alpha/2$ and this angle to be $\beta/2$.

Remember, curvature is 360 minus the sum of the angles there. That means that's how much extra material you need to stick in until that vertex is flat.

So if I stick in an angle of $\alpha/2$ here, at this point, and there's two copies of it. Because there's the front triangle and the back triangle, Then I'll be wedging in α , which is exactly what it takes to make x flat. And this is exactly what it takes to make y flat.

The result will be I'd obliterate x . It's no longer a vertex. I obliterate y . It's no longer a vertex. But I add this new vertex, z . In that way, I remove two vertices, add one, which decreases n . And then I can use induction.

Now, there's lots of things to say why this works. But this is the crazy idea. It's the Alexandrov induction, and it's useful.

If you want to prove, for example, that the star unfolding doesn't overlap, you use this induction also. It's the only proof that's known. So why did we need these properties? Well, the biggest triangle you could make, something like this, like we were doing in witch's hats, where this is almost 90 . This is almost 90 . This is x and y .

The biggest we can make $\alpha/2$ is almost 90 . And the biggest we can make $\beta/2$ is almost 90 . So we'd really like α and β to be less than 180 . But then there is a valid triangle that does this.

Once you fix two angles, it of course determines third one. It might be very close to 0 . So z might have very large curvature, but that's OK.

What else? The other thing is, in what sense could you induct? And this is where I'm going to be a little hand wavy. Yeah, I've found some other polyhedral metric. It has one fewer vertex.

So by induction I know that it's realized by a unique convex polyhedron. The how in the world would I convert that into a realization of the original polyhedron?

And the answer is continuity and magic. The continuity is easy to see. When I glue in this triangle, I don't have to just glue in the triangle with those final angles. I could actually glue in a very tiny triangle initially. And I could change these angles continuously. That makes me feel a little bit better.

I start with something that's very, very close to my polyhedron. Obviously, if I could realize that, I could still realize the original, right? It's almost the same, if I glue in a tiny, tiny sliver of a triangle. That's not obvious, but it's true.

And you vary it until the point, where you actually get something with one fewer vertex. Then you actually do get a realization. And then you continuously morph it back to your original. That's the idea.

I'm not going to detail that idea. It uses something called the inverse function theorem, which is a cool theorem in real analysis. But I think most of you haven't had real analysis. So you can read about it if you're interested. It's cool.

The main consequence for us algorithms people is that it's non constructive. It doesn't actually tell you how to do this. It just says, well, by continuity, it really ought to work out somehow. But we don't know how. We've tried to make it work out and failed.

While this does sort of prove the theorem by continuity, it doesn't actually give us any idea of how to find what polyhedron we are realizing, which is a bit frustrating. And that was the state of the art until four years ago.

Four years ago, these guys, Bobenko and Izvestiev, came up with a constructive version of Alexandrov's Theorem. And I just want to sketch an idea of how this

works. But the end consequence, which is done in this class, I think, as a class project three years ago, is that you get a pseudo polynomial algorithm for solving this problem.

So I'll jump ahead a little bit. Running time, we get-- this, again, for amusement value-- 1891. It's a good year. I think I said, last time, that n to the 125 times r to the 41 was my record. Well, it's been surpassed. I just forgot about it.

n to the 456.5 times r to the 1891, over ϵ ϵ . Here is the desired error. So you can make the error as small as you want. We can't do it perfectly. But your running time increases as the error goes down.

This is our usual feature size. It's the longest length in your input metric, let's say, divided by the shortest length. So that's a measure of how tight the features are. We had that before with carpenter's rule stuff. And n is the number vertices.

Obviously, this is a very bad bound. But the point is it's pseudo polynomial. It's a bad bound for the same reason that our last giant ginormous bound was ginormous. Because differential flow is hard to get good bounds on.

See it's possible to get some kind of bound. But they don't usually represent how fast the algorithm actually is. Again, this algorithm is pretty practical. There's an implementational. I'll show it to you in a moment. Maybe I'll show it to you right now. It'll be more fun.

It is available on the web for free. Try it out. I assume this is a student of Bobenko and Izmestiev in Berlin, and he implemented this program, Java.

So you specify a polyhedral metric. Now, that's the tricky part. And especially, it's tricky here, because it has to be triangulated. But we would normally do it by specifying a polygon plus a gluing. But of course, by computing all shortest paths, you can eventually make that a nice triangulated thing.

So here I have a very simple triangulated thing, namely, a complete graph on four vertices, which makes a regular tetrahedron. It doesn't look so regular, because I

have a lot of perspective in a small, narrow angle of view. But that is a regular tetrahedron, believe me.

Then I can change the edge lengths. Right now, they're all one, so I should be able to make one of the edges longer, like 2. I haven't tried this. We'll see what happens.

I say, Compute, and it says, Starting Calculations-- not my intended answer. I like it to finish the calculation. That's why it prepared some other examples. I suspect it's not the program failing, but I've had some trouble running it on this computer.

I think we have demo effect. Finally, something failed in this class. It took a long time. You'll have to try it online.

You can change the edge lengths to all sorts of crazy things. If you have very close to degenerate situations, it does tend to fail. Because there's some tricky issues to deal with and this floating point.

Oops, it might be doing something. Dare I try to open a file? Here, you get to see my organizational structure or lack thereof. Courses, Slides, 15, and let's take Cube. Exciting.

This is a cube. It's not so obvious that it is a cube. But you set the edge links. And a lot of these are sort of diagonals. And if I do it right, maybe add some iterations. It says, [INAUDIBLE] successfully constructed. You can see it, right?

My job is failing, I'm afraid. Sorry about that. Go away. You'll have to try it at home.

AUDIENCE: It was there.

PROFESSOR: It was there? Oh.

AUDIENCE: [LAUGHTER]

PROFESSOR: Wait, I don't see a cube. What did you see?

AUDIENCE: It was there.

PROFESSOR:

It was there. All right, you saw it. It's like it ghost. It's the left over from Halloween. Let me tell you a little bit about how this works. And you'll have to try the program at home.

A crazy idea is we have this polyhedron, which we want to find. It's a surface. It's hard to think about surfaces that live in 3D. So I want to make it more three dimensional.

And the way I going to do that is by thinking about the inside of the polyhedron. And say, well, imagine it's convex. So there's some point inside the polyhedron. Start with one point. And then, I'd like to fill the interior in between.

So the surface, as you saw-- we're assuming that everything is initially triangulated. We can do that by adding shortest paths. Well, if this triangle is on the surface, and this point is in the center, the natural thing to do is to make tetrahedra inside.

If I could find the 3D triangles, I could also find the 3D tetrahedra. So we're going to do a harder job of finding the 3D tetrahedra. From that, we will derive what the triangles are-- easy enough idea. Make your problem harder. That often makes things easier.

How do we do that? Well, for every vertex up here-- call it v_i -- there's some distance. This dashed line-- we're going to call that r_i . That's really what we'd like to figure out.

We already know the triangles-- we know their geometry. That's given to us as part of the metric. We don't know how the triangles are hinged, with respect to each other. That's the dihedral angles. That's the tricky part.

But we're not going to worry about that directly. We're going to think about these edge lengths, r_i . Because if I knew all the r_i lengths-- they're the sort of radii from the center point-- then I'd know all the edge lengths of the tetrahedron that uniquely determined the tetrahedron.

If I do that for all the tetrahedron, there would be a unique way to glue them

together. And I would get the answer. So I really just need these r_i 's.

What I'd like to do is characterize, when are the r_i 's valid? And there's a few conditions to make them reasonable. The first thing, I guess I didn't even write in the notes. You might call it sanity, which is that these are real 3D tetrahedra.

I'm not going to try to characterize when that's true. It's some function on the r_i lengths and existing lengths. But I always want these to correspond actual tetrahedra. The trouble is, they might not fit together right to make 3D polyhedron. So I won't really have to worry about this condition.

The next condition-- what do I call it-- is convexity. This will also be kind of easy. What I want is that the outside surface is convex. Now, if you look at the outside surface, what we really want is that one of these edges, on the outside surface, bends the correct way.

So when will that be true? Well, I'm going to look at the two triangles next to that edge. That defines two tetrahedra that share a face. And if I knew what the r_i 's where, I'd know what those tetrahedra are.

And what I really need is that if I look at the angle-- I'm going to draw this a little bit thicker-- this dihedral angle between-- it's hard to draw. If you look at this edge, there's two tetrahedra glued there.

So there's some solid dihedral angle of tetrahedron glued at that edge from this tetrahedron. And there's some angle over here that's glued to this edge from that tetrahedron. I want the sum of those two dihedral angles, the total material there, to be less than or equal to 180.

That is, the sum of those angles is really the dihedral angle along this edge. If it's less than 180, it'll be convex. It's hard to draw, but very intuitive.

This is also pretty easy to get. You can set the r_i so that that's true. The hard part is looking at a vertex.

You think about a vertex. It has a whole bunch of triangles around it. From this vertex to the central point, which we usually call p , if you look at that edge-- I'm not going to try to draw it-- there's, in this case, five tetrahedra glued around that edge.

The total dihedral angle of those five tetrahedra ought to be exactly 360. If these tetrahedra are going to fit together and make a solid thing-- and this we call the reality constraint. I'm not usually bound by the reality constraint. In fact, we will relax it a little bit.

But we would like that the sum of the dihedral angles around some edge from P to v_i equals 360. This is the hard constraint to get. Reality is tough. Sanity, convexity, easy.

I guess I shouldn't kill this yet. So let me tell you a little bit more about how we get this.

I'll just tell you some of this stuff. The first thing we do is get sanity and convexity and forget about reality. Reality is overrated. To do that, we actually need two things. One, is we need a triangulation of the surface. We're presumably given some triangulation, but that one might be hard to work with.

And the answer is, to start out, it's nice to use Delaunay triangulation. How many people know about the Delaunay triangulations? They're nice triangulations. And you can define Delaunay triangulation in an implicit surface like this.

I won't do that here. But it can be done, and it can be computed. One of the innovations in this paper was to actually show how to compute it in pseudo polynomial time. It's not too hard. It's just shortest paths.

Then, it turns out, if you start with that triangulation, and you let all the r_i 's be really, really big, close to infinity, then sanity and convexity hold. So I guess intuition may be like witch's hats. I don't know. You're making these super-sharp triangles, tetrahedra. And it will work.

Now, it'll make everything big and equal. And so reality will be violated. But sanity

and convexity are OK, and that will be your starting point. So now the crazy idea is to fix it.

I'm going to define a kind of curvature around each edge, around that center line from p to v_i . That's the sum that we wanted to be equal to 360. And now I'm just taking 360 minus that. That's the deficit. This is how much we're off by.

So we have some starting point. And it turns out the K_i will be positive, I believe. And we want to make K_i evolve with time, so that it is $1 - t$, times the initial K_i .

So we start at times 0, with the solution where all the r_i 's are large. So we're staying in convex, and we're going to be staying in convex from then on.

And then, if we could do this, we would start at t equals 0. This is 1 times the original curvature at every edge, so it's a problem. But at t equals 1, at time 1, when this is finished, we will get 0 times whatever we had. So all of these numbers will magically go to 0. And then it adds up to 360.

It's not obvious that this works, but it does. It defines a differential flow. And those who know differential equations, this will look pretty.

This is partial differential equations to make it a little more exciting. This is a κ and inverse and ϕ . What was ϕ ? I'm forgetting my own notation, some initial value. It should probably be r . r ? We'll see.

All this is saying is what I really want is to evolve κ 's to go smaller. I don't know how to do that, because all I can do is change the r_i 's. That's the only thing I control. The κ 's are computed as a function of the r_i 's.

There's a standard trick in analysis-- or I guess, whatever that is, calculus-- to rewrite things. You get to control the r_i 's. You want to control the κ 's.

So you use this thing, which is called the Jacobian. It's a matrix. But really, it says, if I change r_i , how does that change κ_j ? It says that for all i and j . So it's just a whole bunch of different values. They're really functions.

You invert that because you really want to know how-- I want to change the r_i 's in order to affect the $kappa$ s. So you write this differential equation. And you just follow your nose, and it tells you what to do. I'm going to slide a whole bunch of details under the rug.

The hard part is proving that this relation is actually feasible, that you can change the r_i 's in order to change the $kappa$ s, no matter how you want. This is equivalent to saying that the Jacobian is never 0, and it always has an inverse. And that's hard to prove.

And in fact, they prove it using the Inverse Function Theorem, which is what we were trying to avoid. But hey, it's a construction of sorts, and it leads to that implementation. This, in some sense, gives you an algorithm.

To turn it into an actual algorithm-- as we did in this paper, which appeared last year-- you need to make this proof a little bit more constructive, a little bit more algorithmic, let's say. You don't just want to prove that this Jacobian is never zero 0 and that it always has an inverse.

You need to prove that it's actually always at least some epsilon. It's bounded away from 0, and same for the inverse. Instead of using inverse functions there, and we're totally constructive, we just look at second derivatives and bound those.

We get a really awful bound, which is this one. But we do get an explicit bound on how long, essentially, the algorithm is implemented and the applet takes. So, a high-level sketch, but it gives you an idea. This is pretty new stuff. Questions?

I want to move on to more fun topics, let's technical, I should say. So that's Alexandrov's Theorem algorithms. In theory, there's now a computer program that will tell you how to fold something, as long as we have an Alexandrov gluing.

So it has to be polyhedral. That's pretty much for free. It has to be convex. That's the interesting part. It has to be a topological sphere that is just non-crossing.

So let's think about gluings. We know that as long as we find a good gluing, it

corresponds to something. We know that something's a little hard to find. But we'll see how we find them in practice. It's kind of fun.

We talked about the decision problem. Is it possible at all for a given polygon? So here's a polygon where it's not possible at all. It's kind of fun-- ungluable. I should make that sharper.

So this is a polygon. This is the material. I want to glue it into a convex shape. I claim it's not possible. The reason is, think about this vertex. What could get glued to that vertex?

It has to be less than 360. Right now, it's less than 360. It's that much material. Could I glue any other vertex to that point? No. Because all the other vertices have at least 90 degrees of material.

This one has even more. But all them have at least 90. And there's less than 90 remaining. So no other vertex could be glued there.

The other possibility, it doesn't have to be a vertex. It could be a point in the middle of an edge. But that has 180 degrees. That's way too much, so nothing fits there. Therefore, I have glue like this.

This is what we call zipping. I zip those two edges to each other. Now, they happen to be exactly the same length. Therefore, this vertex gets glued to that vertex.

But that's not allowed, because there's more than 180 and more than 180 here-- contradiction. So that's it. This can actually happen a lot. Let me formalize that.

So we could look at what you might call the generic case. And I won't call it the generic case. But there's a back reference there, all the linkage stuff. I can explicitly think about the random case.

Suppose you choose all the lengths of your edges and all your angles at random in your polygon. You get some messy thing. But the point is, around half the vertices will be reflex angle, more than 180. And those are kind of troubled.

Because if you have a reflex vertex like this, in particular, you can't glue the middle of an edge. Because that has 180. You can only glue convex vertices that have some smaller angle into there. Otherwise, use zip, and zipping would be like this.

So, what do I want to say? Yeah, I see. Right, cool. Before I get to zipping, maybe I glue some convex angle in there. It's the same kind of analysis. Maybe I glue some convex angle.

But if the angles are chosen at random, I'm not going to get lucky and completely flatten out the vertex. It won't disappear. That would be too much to hope for.

So most the time, I'm going to glue in-- maybe I glue in some material. Then I glue in some more. At some point, I get stuck. I can't glue anymore in, or I decide I don't want to.

At that point, I must zip. This is a rather ambiguous notation. But zipping, I want to glue this edge to that edge. Well, how long can I zip them for?

Well, let's say I zip them. At any moment here, I'm gluing a 180 degree angle to another 180 degree angle. I can't stop zipping once I start. Because there's nothing else that could fit in there. So I've got to zip until I get to a corner.

There's going to be some corner, some vertex, that's glued to some point. Now, is it going to be glued to a vertex? Well, that would be awfully lucky. That would mean that these two edges have exactly the same length. That's not going to happen randomly.

One of these is going to be larger than the other. Now I'm gluing some angle to a 180 degree angle. The result is another reflex vertex. This will keep going. And you'll never finish.

So this is a bit hand wavy. But you can see the textbook for more formal argument. I have it here. I'll finish it for you. This would be fine if this vertex is convex.

This is less than 180. But actually, with probability $1/2$, it's going to be a reflex vertex. And then you're screwed, because now you have more than 180 gluing to

180. That's more than 360.

So you don't even have to go all the way to the end. Immediately, after the very first zip, with probability $1/2$, you'll fail. If you have n vertices, and half of them are reflex, a quarter of them are going to fail.

So that's plenty, as long as n is bigger than 4. For n large, you're going to get exponentially high probability that this fails. Most of the time, if you're choosing random polyhedra, random polygons, you can't glue them.

So you've got to be clever, choose some nice polygons. It's actually not that hard. Like the cross, if you just glued together a bunch of unit squares, it'll go into something nice probably. Or here's another great, class. Convex polygons.

Convex polygons do always glue to something using something called perimeter having. What to do, basically, by having the perimeter.

I have some convex polygon. I'm going to choose any point on the boundary. And then have the perimeter from there. So I walk equal distance, all around this thing-- equal speed, I should say. Something like that.

This is x . This is the antipodal point, in terms of measuring perimeter. Now I zip. So I'm going to glue this to this. I'm going to glue this to this. I drew all those marks. I was planning ahead. Can you tell?

I was trying to make all the segments the same length, so that I could properly glue. Something I've never really said about gluings, but you should always glue same length things to other things. Segments.

OK, there we go. There's a gluing. There's no crossings in that gluing. You can check. It's Alexandrov. It is convex, because what do I glue to what? Well, here, maybe I'm going this vertex to the middle of this edge. But it's a convex vertex. So that's OK.

Everything looks like that. You're always gluing some vertex to some other edge.

And all the vertices are convex. So it will be less than 360. Maybe, if you got lucky, some edge will glue to some edge. But that's still barely OK.

So for convex polygons, it's always going to work. Let's try it out, shall we? I don't want to use this page, because I'm not done with it. We'll take page one, make a convex polygon.

I don't want to make it too complicated. I'll spend a lot of time taping. You can try this at home with your own, favorite convex polygon. Here's a convex polygon. And now, we're going to pick a point. Do I have a good point written here? Let's try here. See that point, purple?

Now we are going to zip. So take this. I glue those edges together. I don't have glue, or glue takes a while to dry. So we're going to use tape. Ta-da.

Now, we have to glue a little more. I'm going to pretend those just met. Eh, it's a little ugly. Man, couldn't I have chosen something a little more degenerate? Here, it really does work in the random case, which is fine, as long as I was restricted to convex.

Now, we're going to If you brought tape, you can try yourself. Tape a little more. See, it's already making a convex polyhedron. It's like magic. For a long time, we've tried to turn this process into an algorithm. Because it looks pretty good, right?

But we have not been able to formalize this. This differential flow does not really follow our intuition. Now, I'm going to glue these are edges together. This is when I wish I had three hands. A little more.

And there, we've computed the antipodal point. In theory, you would zip that. But I'm going to leave it open to give myself some maneuverability here. So you puff it out a little. It will be kind of round and convex.

Now, that's not quite a polyhedron. But this is a polyhedral metric. And Alexandrov's Theorem says this should make a unique convex polyhedron. And with a little bit of practice and thinking about it, you can derive that actually this must be an edge

here.

It's more of an intuition thing. See, it looks like this really wants to bend there. So that must be an edge. This one looks like it must be an edge here. This gets a little hard to do.

But this is how we used to do them. Wow, this probably still is how we do them. I haven't done a lot of these in a while. It looks like there's an edge here. Oh boy, something like that.

Here there already is an edge, kind of convenient. Here there's an edge. I'm just folding point to point. I'm almost triangulated. I've got a nice line there.

It's starting to look like a polyhedron. There's a couple missing. This guy probably has a diagonal somewhere. I would guess here. And on the bottom, there's a diagonal missing, I would guess, here.

One of them is going to make it reflex when the other one will make it convex. You can experiment. Then you unfold, ideally, and make a better version of this with more properly drawn lines and then refold it. And I bet you it makes a beautiful convex polyhedron.

So this is the experimental approach. And this gives you a hands-on feel for Alexandrov's Theorem. Like that. Yeah, almost.

Believe me, it's a triangulated convex polyhedron. Gorgeous. That's how it works. And you can try for your own convex polyhedron where you're guaranteed not to fail.

Now, you prove some nice things about these perimeter havings. In particular, I had this flexibility of where I started and where I ended. And I could move those points continuously. There's infinitely many choices for x and y .

And certainly, there's infinitely many gluings. I can just keep moving x and moving y correspondingly. There's a whole continuum of gluings. They're all different.

They all generate different polyhedra, if you think about it. Because if you think of x being very close. x is definitely going to be a vertex of your polyhedron. Because you end up gluing this part of the edge to this part of the edge. So you have 180 degrees of material there. That's a vertex, less than 360.

And this is also a vertex, let's say. Look at the nearest neighbor to x . Here is a distance between two vertices. It's the shortest path. If you make x very close to the vertex, it will be a shortest path.

And you're setting that to infinitely many different values, a whole continuum, very close to 0. If you look at some polyhedron that this thing realizes, it has whatever, n vertices. You look at the n squared different shortest paths between them. That's some finite set of real numbers.

And here, I'm supposed to get a continuum of real numbers. Therefore, the set of all polyhedra that you are making in this process is a continuum. There's a lot of them.

Most of the things you'll get out of this are different polyhedra. If I started with a different x and y , I'd get a slightly different polyhedron. It will change continuously, but it will be different.

So this is the sense in which, at the beginning, I said there infinitely many gluings. Really, the only time that they're infinite is when you have a convex polygon somewhere embedded inside your polygon. It doesn't have to be at the top. All right, where do I want to go? Here.

One way to think about gluings is all these arrows that are not crossing. But it's a little bit awkward to work with. And next class we're going to really find algorithms to compute these gluings.

And it's useful to have another structure, another way of looking and gluings, called the gluing tree. And the main idea is to take your polygon, which I'm going to explicitly shade. Because you're soon going to lose track of where it is if I don't shade it.

So that's the inside of the polygon, and there's outside space. I want to turn this whole picture inside out. Think of this as living on a sphere. On the sphere, there's some shaded region. That's the material. And then there's the rest, the black of the blackboard.

If I look on the other side of the sphere, my perspective will be, I have some convex polygon. And the interior of the polygon is on the outside. This is a useful way of thinking about things.

So there's my polygons out there. Now in fact, in a convex case, this would be less than 180. It's hard to see that in this picture. But just think of it topologically. Don't worry so much about the geometry.

Now, what I'm trying to do-- it's a little easier to think about, oh, I need to glue this edge to this edge and then maybe glue up to whatever. Maybe these were the full length. Then I'll glue this edge to this portion of that edge.

So I still can draw my arrows, something like that. This would be a parameter having gluing. I can still draw them, but now they're on the inside instead of the outside. It's easier to see them. And in particular, I could actually say, well, if I want to go that edge to that edge, why don't I draw them next to each other?

So it would look like this. Then have these edges when I draw them next to each other. And then I have those edges and then that edge and then the little edges, except these are actually the same point. So it would look like that.

I think of these as almost touching and then the material still on the outside. It's not looking so white anymore, but you get the idea. Well, that's a path. In general, you get a tree.

So a gluing tree will look something like this, maybe. It's a double tree. There's an underlying tree here, and then you walk around it. And the material is all out here. It's like that's where you're gluing.

Now, this is actually really intuitive to us. It could be. When we were thinking about where to cut along the surface of a convex polyhedron, at some point we said, well, in a convex polyhedron, we had all these properties that, in the end, said the cuts must be a spanning tree of the vertices.

Every vertex must get cut too. And you can't have any cycles in the cuts, because that would disconnect the surface. So where you cut was a set of was a tree, a set of edges forming a tree.

That's the cut tree, just viewed from the other side. That's where we want to glue. Folding is supposed to be exactly the opposite of unfolding. That's why the gluing tree is a tree.

You could think about some interesting things like where are the vertices of my polygon in that picture? So here, I made all these notches, but not all of them were vertices. Some of them were the middles of edges.

So this one might be 180, meaning it wasn't really a vertex of the polygon. It was the middle of an edge. Obviously, if I look at some vertex where three corners are coming together, if one of them was 180 and another one was 180, we would not have room for a third.

At a degree 3 vertex or more, it could have, at most, one middle of an edge. These guys, the other remaining guys, must be vertices of the polygon. The middle of an edge could be two 180s. That's actually the typical case.

If you look at one of these gluings, most of the points, it's the middle of an edge gluing to another middle of an edge. That's what's happening almost everywhere. But you could also have some 180 gluing to some vertex. But then it had better be a convex vertex, less than 180.

You can go around and think about what all these situations are. When there's a 180, the remaining guys had better be convex. And their total angle had better be less than 180.

What about the leaves? This is a zipping place. I was talking about zips before, were you just locally join the two incident edges to themselves. The leaves are where zips happen. Trees have leaves, so zips happen. Zip happens.

[LAUGHTER]

PROFESSOR: This angle, obviously, will be less than 360. It could be a vertex, and then at some angle, or it could be the middle of an edge. And then it will be 180.

That's what we had over here. I chose one and x to be a vertex. But then, generically, on the other side, you're going to get the middle of an edge. And the example I did in reality, both of them were in the middle of an edge.

I'm going to make special note of the ones where we have material like this, and you end up zipping like that, we call this a fold point. I don't know why, but you fold at it. So it's some shorthand.

So these are fold points, the leaves that are at the middle of an edge, they're a little special or a little restricted. Because how much curvature is at that point? 180. Fold points have curvature 180. The total curvature is 720. So I can't have more than four fold points.

They're actually quite rare, but they can happen. Obviously, we had two of them in these examples.

Can actually have four? Yeah I think.

AUDIENCE: Could you have [INAUDIBLE]?

PROFESSOR: I'm going to draw an actual folding, with creases and everything. This is a blintz space, for the origamists. You fold like that. You get a doubly covered square. The gluing tree is going to look like this. You see that?

And it will be drawn right on top of that, and the actual material is back here. So this will be exactly four fold points. And just by luck, you have these four 90 degree angles. So there's no more curvature there. So yeah, you can definitely get four fold

points. I don't know if you can get a large examples four fold points.

AUDIENCE: I think four fold points means it'll be flat.

PROFESSOR: Four fold points means it'll be flat.

AUDIENCE: So that's going to be 0 curvature at that.

PROFESSOR: You're going to have 0 curvature at every other vertex. It's certainly implies that you are a tetrahedron. Because you only have four vertices at all. So yeah.

You might be able to start with some complicate a polygon. And then, everything somehow disappears, because all the angles match up in a lucky way, like this. But the polyhedron itself will definitely be a tetrahedron.

I'm not sure if it has to be flat, but tetrahedron is pretty simple. So it's not so exciting.

Let me tell you about rolling belts, because they're really cool. And then I think we're done.

A rolling belt is basically an embedded, convex polygon. And we saw that when we looked at a convex polygon, from the perspective of a gluing tree, it was a path. In general, I have a tree. And I want to define when a path in that tree is like a convex polygon.

And like in the sense that it can roll. I could move x continuously and move y corresponding, and I'd always have an Alexandrov gluing. So it's like one of these infinite families. I'd like to detect those things, and we call them rolling belts.

They're going to be a path in the gluing tree. Here are the conditions you need. The endpoints are either fold points or convex vertices, no reflex. In that case, they must be leaves. And along that path, you always have less than or equal to 180 degrees of material on either side.

This is pretty intuitive. So you have some tree. I'll draw a slightly more general

picture here. And then, I look at some path in that tree, maybe from here to here.

Yeah, this was the way I glued things up. Now, imagine that I recut here. Ideally, these are supposed to be glued together. But what if I separated, I pulled just those parts apart? I left these intact, all the sub trees hanging off I didn't touch.

And then I just rolled that thing like it was a convex polygon. Which means, instead of going here to here, I'm going to glue this point to this point. I'm going to slide this over a little bit.

Whereas before, this point was glued to itself-- that was the end of the zip-- the new zip point is going to be-- I can't tell, everything's moving this way. This point is going to move up here. And it's going to become the new zip point.

Everybody here is moving in that direction. Everybody here is moving in that direction. I just kind of roll that belt along, and then I reglue.

And as long as it looks like a convex polygon-- meaning, as you walk around here, the total amount of material getting glued, like, here I have two portions of material-- as long as that's, at most, 180 in total, I can do that.

Because if it's, at most, 180 on both sides, it will be, at most, 360 in total. So that's it. And there's a separate corner condition. But all it's saying is that it's, at most, 180 everywhere. And then it's really acting like a convex polygon, but not in total, just in that little subpath of the tree.

Turns out, this is the only way to have an infinite set of gluings. All infinite gluings will look like rolling belts. Now, there can be more than one rolling belt in a single example. And this I can even build.

It's pretty simple. I take a rectangle, and I glue it into a cylinder. Beautiful, what an amazing cylinder. But I'm not going to commit to how the rest of the thing is glued.

There's a cylinder. What did I draw? The gluing tree. So initially, my rectangles out here, imagine. And I've glued-- what did I glue, the shortage to the shortage?

So what I end up with, in terms of gluing tree, is I have these two edges that are glued to each other. And then I have this big thing, which I don't know how I want to glue that yet. I haven't decided.

So those loops are supposed to get squashed into a tree. And that's this loop and this loop. You can almost see this is a weird projection of a cylinder. A cylinder goes around on the backside, because all the material is out here.

I'm totally free, what I do with the top here. It's a rolling belt. Look, I can even roll it. It's great. I can slide it around, slide, slide, slide. I can do this forever, because it's cyclic.

And then separately, I can slide this one however I want. It doesn't matter. At all times, I'm just gluing 180 to 180. And at the ends, I'm getting to just single 180's. Those are fold points. Well, this one's not technically a fold point if I did it right at the vertex, but it acts the same way.

So the gluing tree is always going to look something like this. But each of these could roll. So it could look like that. It could look like it's even. You get to slide this around. So it's a double continuum, a double rainbow.

In some cases, like this one, when there are lines, you actually get a flat thing, which is kind of boring. In all other cases, you will get a convex tetrahedron, non-flat, for this one example.

That's two rolling belts. Now notice, to have two rolling belts, I basically had to have four fold points. Sounds familiar. I can only have four fold points. So it seems like maybe two rolling belts is the most I can have. Because each rolling belt-- if you roll a little bit, it will have a fold point on either end.

Two is kind of the max, but only kind of. You can do three. Let's do three. Three rolling belts.

I'm going to take this example, the blintz space. Yeah, let's look at this example for a little bit. I hadn't planned this. The tree here, is this nice four star, four-armed star.

And there's kind of a rolling belt from here to here.

If I ignore the fact that there are these things hanging off, then I have, at most, 180 on both sides. It acts just like a rolling belt. Also, there's a rolling belt here. They cross each other. Crazy.

But in theory, it's a little tricky. They're not independent anymore. I can roll one of them however I want. But if I roll that one at all, I cannot roll the other one.

Because as soon as you roll-- let's say I roll the horizontal one. It'll look like this. They'll be if you roll that. So these top and bottom guys are no longer touching. And if you try to make this path a rolling belt, it doesn't look so good.

Because here, this side, there's more than 180. It's going to have a 90 from here and 180 from down there. So that's no longer a rolling belt. Right now it is, and I could roll just the vertical one or the horizontal one but not both.

But the point here is, you can have rolling belts that cross or overlap each other, in general. If we do the same thing with a triangle-- in fact, I think I'm going to make a tetrahedron-- look at that. Amazing. Bring those three points together.

The gluing tree looks like that. This is the top of the tetrahedron. These are the other three vertices of the tetrahedron. It's a spanning tree of the vertices.

I claim this is a rolling belt. And therefore, so is this and so is this. So there are three of them. Because at this point, well, that was like a fold point. There's only 180 degrees there. As I walk around, here I have two 60s going together. That's less than 180.

And on the other side, I just have 160. So it's definitely less than 180 on both sides, all the way along this curve. And by symmetry, along all three. So I can kind of get three rolling belts.

They're not independent, I don't think. If I roll one of them, it looks like actually maybe I can keep rolling the others. But this is the only way you can get three rolling

belts. It turns out, four rolling belts are impossible.

This is actually something we proved in this class. We've proved it six years ago. So it's probably even in the textbook. I don't remember, very simple proof.

If you have four rolling belts, even if they overlap, you have to have a lot of fold points, at least four of them. There has to be at least one fold point per rolling belt, because they have different ends. At least one end is different.

Here, we had three fold points and three belts. If we have four fold points, at least four belts. If you have four rolling belts, at least four fold points. Get the right direction. We know there's only four fold points. So it's actually exactly four fold points.

But as you were saying, when you have four fold points, there's really no more curvature to go around. In fact, this can't happen. Because your four fold points-- all the curvature is at those four leaves. Every other vertex must be perfectly balanced, has to have 0 curvature.

And then things break. It's not a very convincing argument.

You imagine these fold points. They're somewhere in your gluing tree. Maybe they're here. In order to fit four rolling belts, there's got to be, on average, two rolling belts coming out of every vertex.

And where they diverge, it's a problem. Because there has to be basically 0 degrees of material out here for them to roll that way and to roll that way and have 0 curvature. So it fails.

One rolling belt, two rolling belts, three rolling belts, no more. So it's not that complicated, in terms of the infinite structure. What's really interesting is all the very combinatorially different trees. When I roll a belt, not much changes. It's just changing a few of the lengths.

Next class, we're going to look at how the trees themselves can actually change combinatorially and get bounds on how many there are or how to compute them.

There's good algorithms for it, all that stuff. Still some open problems too.

That'll be next class, which is on Monday. No class on Wednesday. That's it.