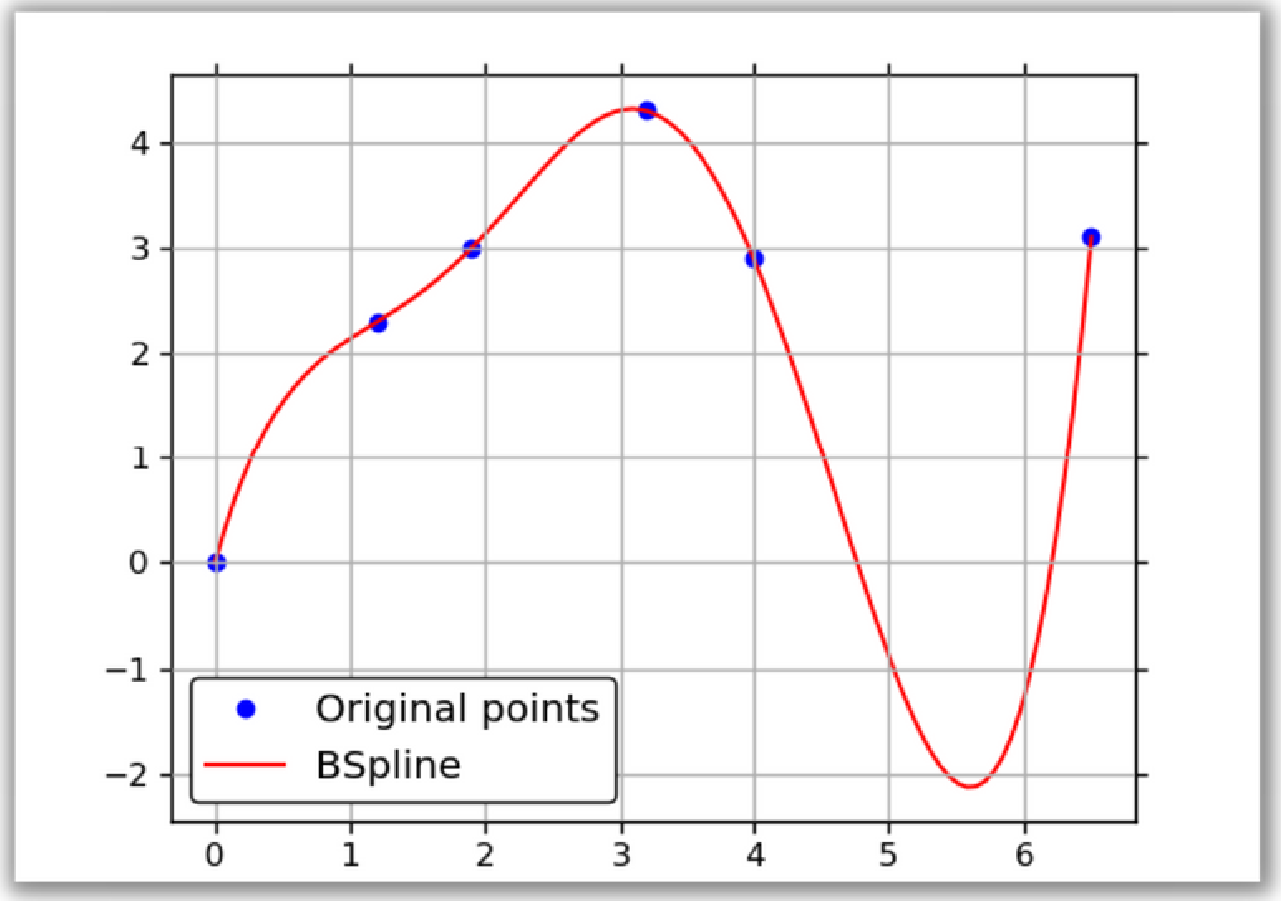
ME F491 SPECIAL PROJECT



bspline surface approximation of point clouds

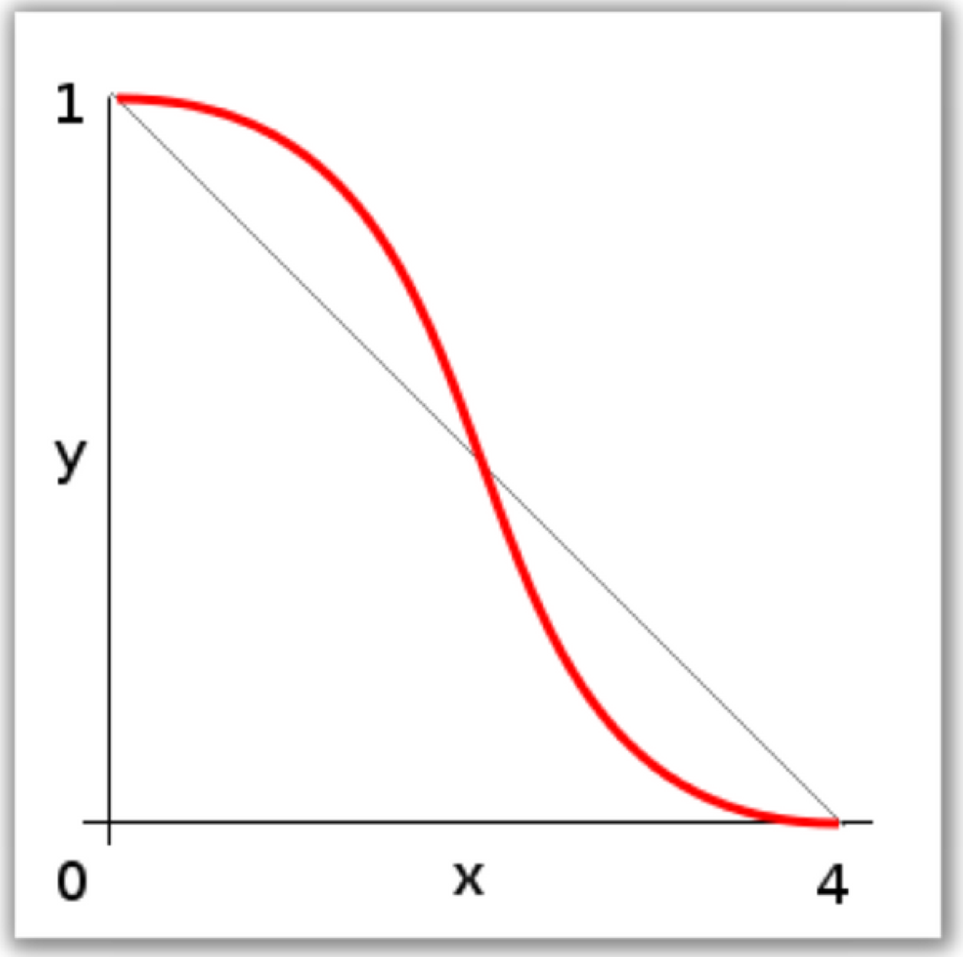
Deven Paul
2018A4PS0047G

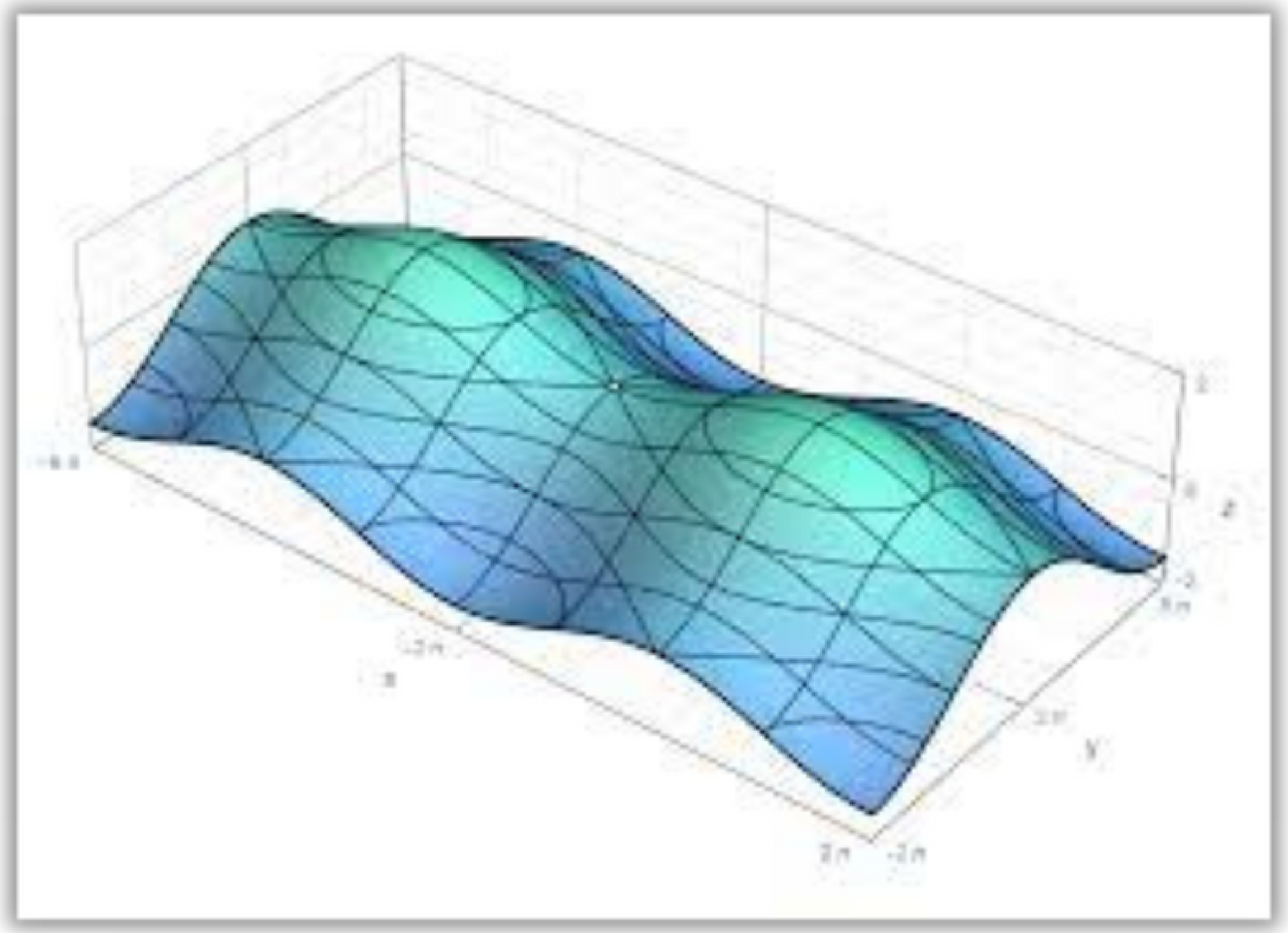
Curves & Surfaces

- In mathematics, curve is an object which is similar to a line but not necessarily straight.

- A curve is the image of an interval to a topological space by a continuous function. In some contexts, the function that defines the curve is called a parametrization, and the curve is a parametric curve.

- In mathematics, a surface is generalization of plane, which need not be flat and whose curvature is not necessarily zero.





Linear Segment

- In geometry line segment is a part of line bounded by two end points and contains all the points on line between these endpoints.

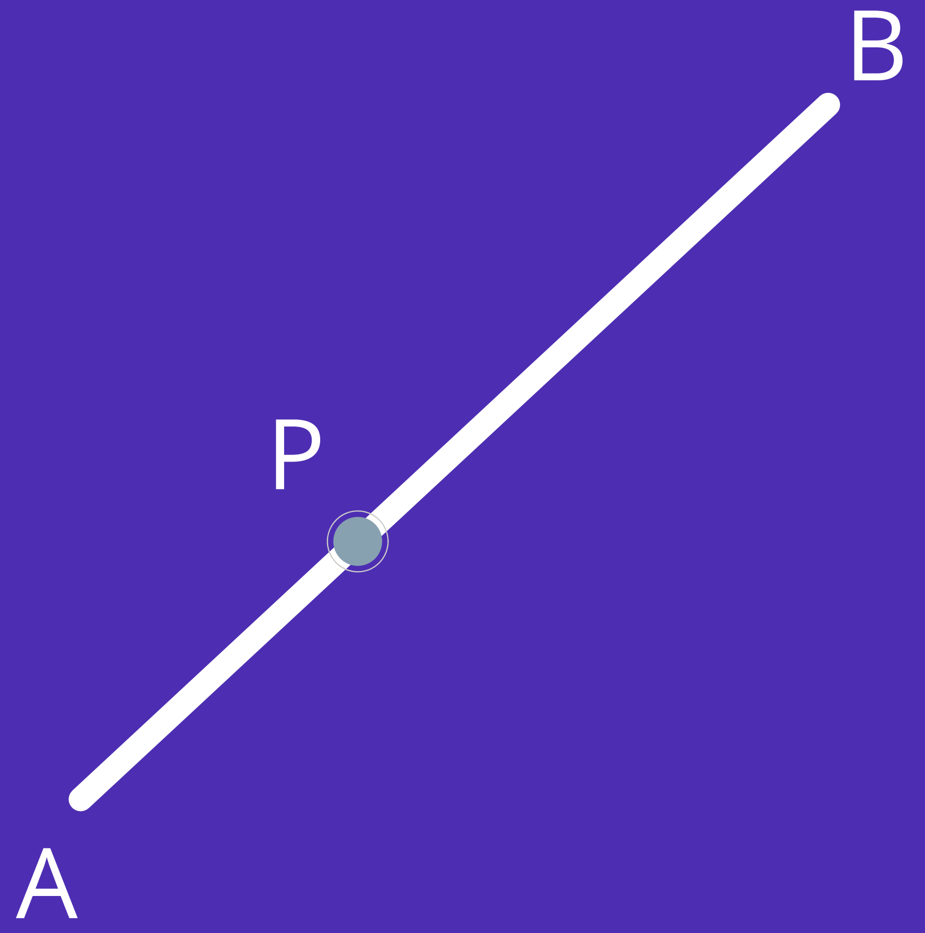
- Progress is term defined to determine the length progress of segment between these two endpoints. Progress is denoted by 't'.

Any point 'P' on this line segment can be determined by:

$$P = A \times (1-t) + B \times t$$

We can break the equation for cartesian 2D planes:

$$PX = AX \times (1-t) + BX \times t$$
$$PY = AY \times (1-t) + BY \times t$$



Linear Segment (contd.)

We can trace a surface using two line segments and similarly we can determine any point on this surface using control points determination on these two line segments.

Let's say that the left line segment(blue) extends from A to C and the right line segment(blue) extends from B to D:

$$P0 = A \times (1-t) + C \times t$$

$$P1 = B \times (1-t) + D \times t$$

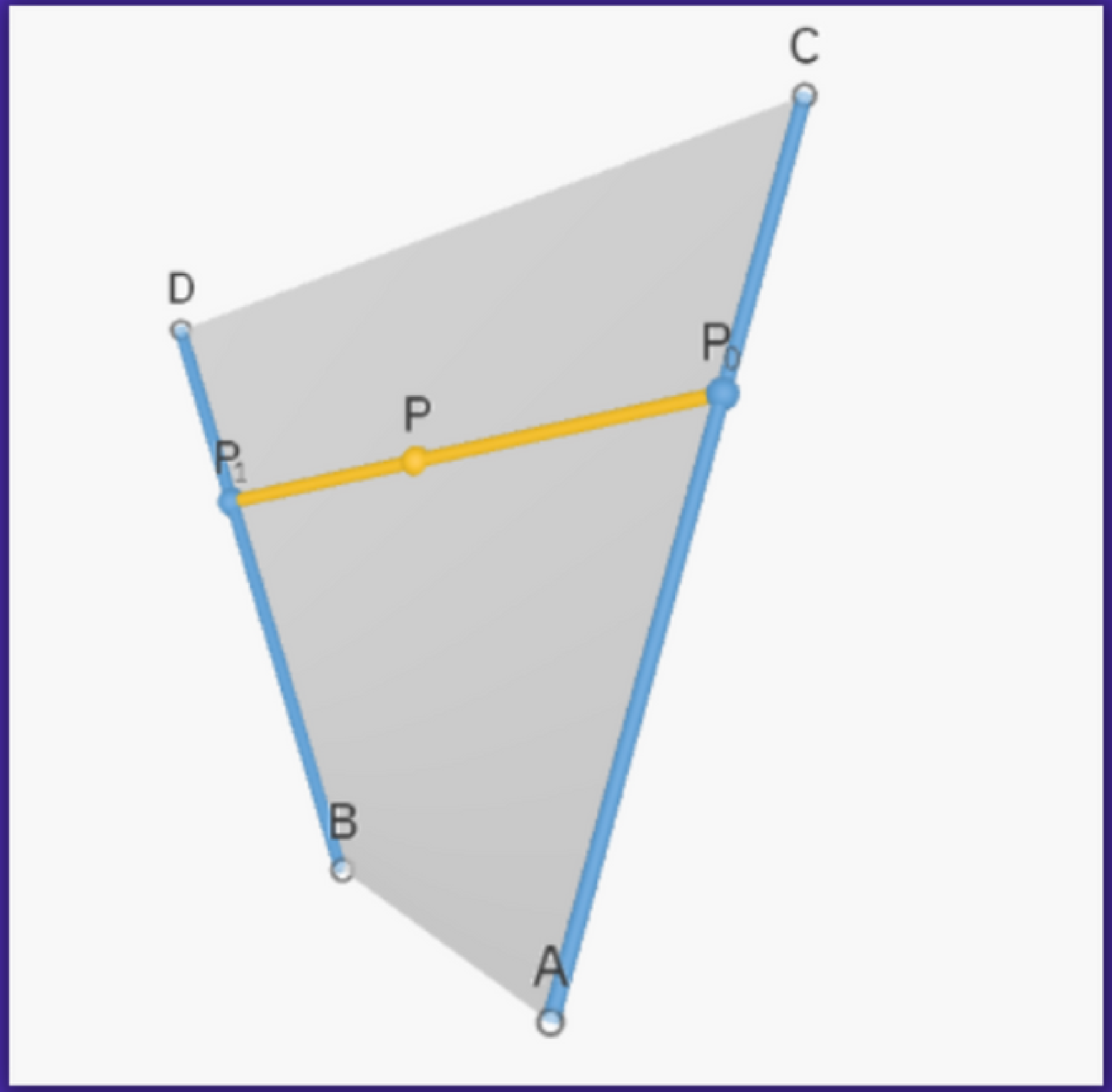
Now the points on the line segment(yellow) is another interpolation (s is the progress on yellow line segment):

$$P = P0 \times (1-s) + P1 \times s$$

$$P = \{ A \times (1-t) + C \times t \} \times (1-s) + \{ B \times (1-t) + D \times t \} \times s$$

$$\Rightarrow$$

$$P = \{ A \times (1-t) \times (1-s) \} + \{ B \times t \times (1-s) \} + \{ C \times (1-t) \times s \} + \{ D \times t \times s \}$$



Quadratic Bezier Curves

- Progress are locked together , 's' and 't' are the same:

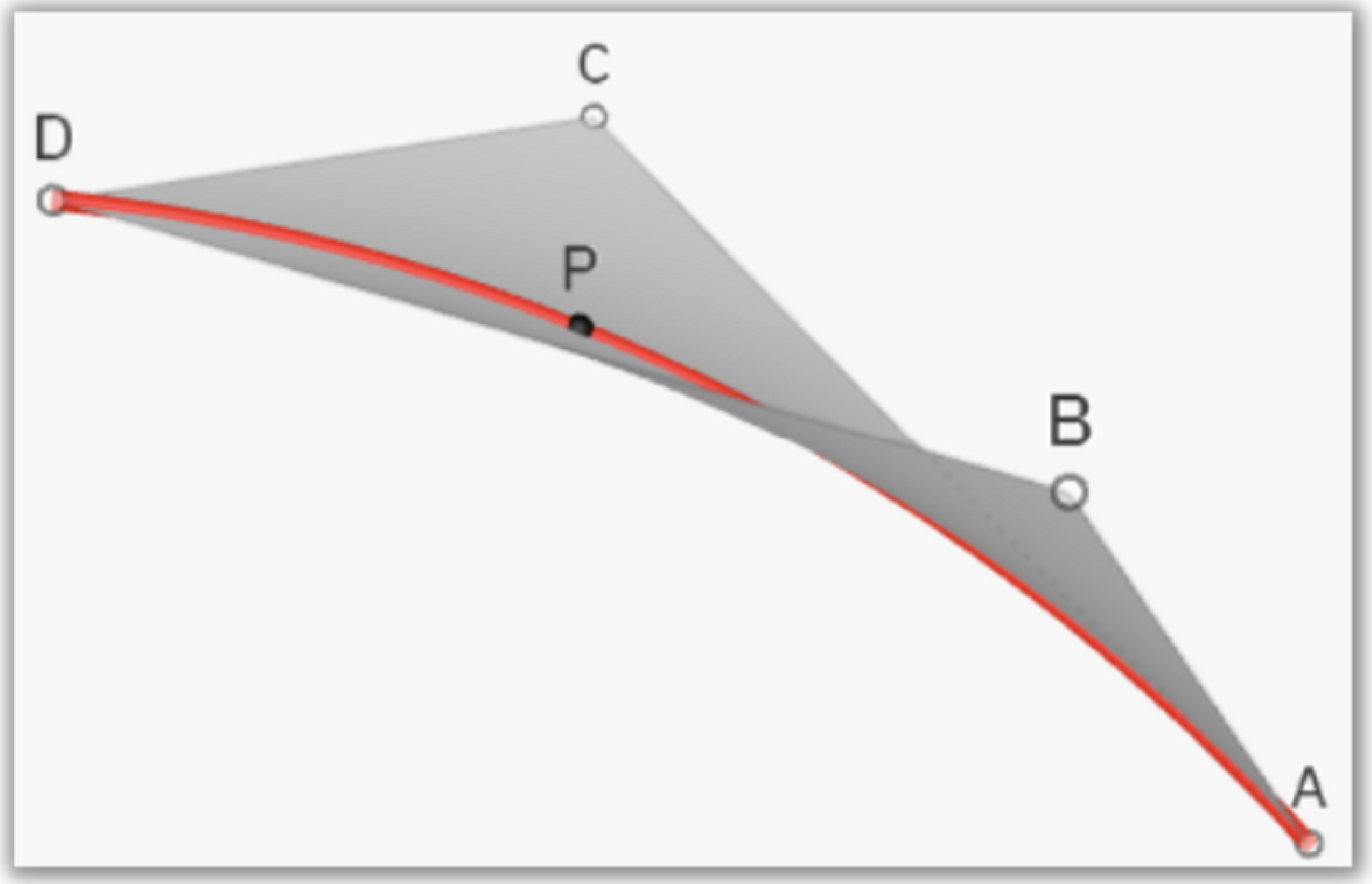
  P = { A x (1-t) x (1-t) } +
  { B x t x (1- t) } +
  { C x (1-t) x t } +
  { D x t x t }

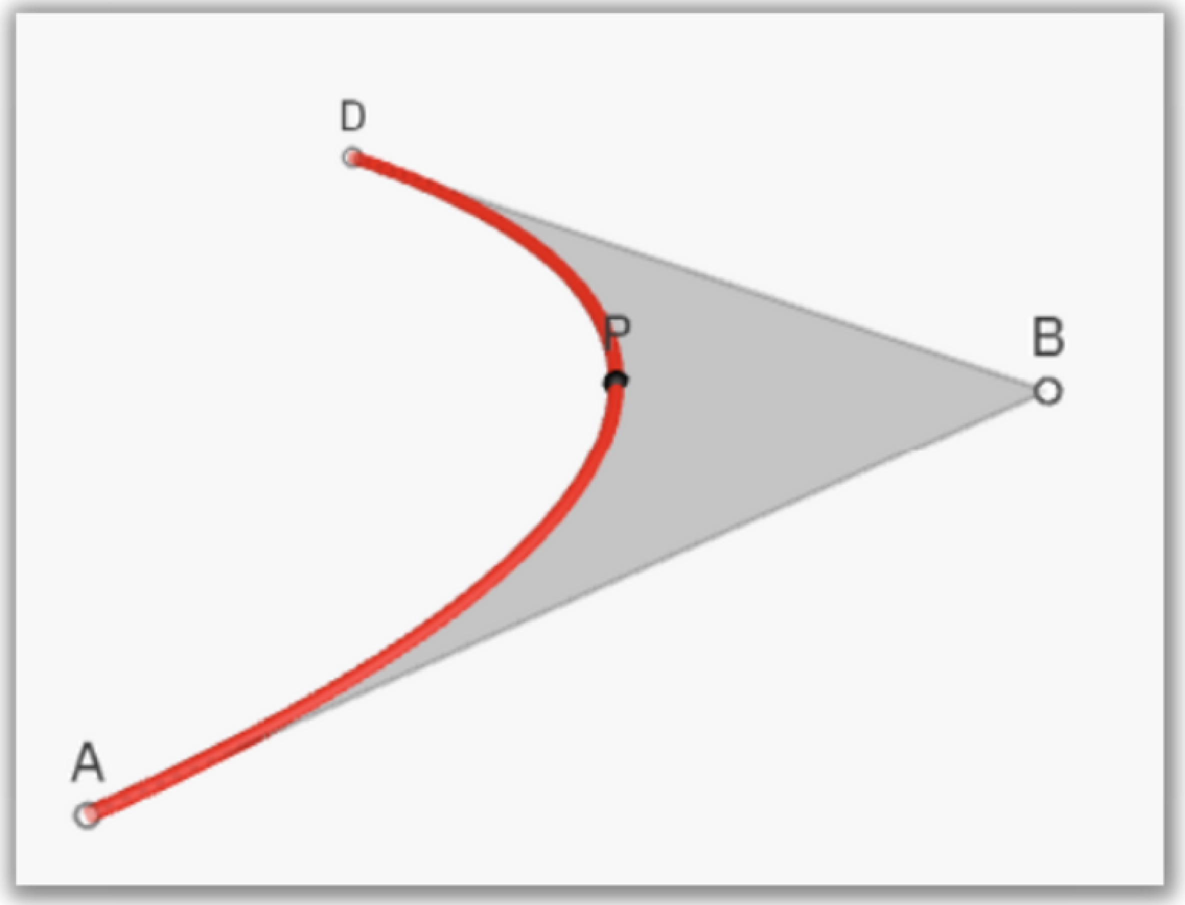
- Both 'B' and 'C' points have same weights:

  P = { A x (1-t) x (1-t) } +
  { (B + C) x t x (1- t) } +
  { D x t x t }

- If we overlap points B & C, we get a nice flat 2D curve:

  P = { A x (1-t) x (1-t) } +
  { 2B x t x (1- t) } +
  { D x t x t }





Cubic Bezier Curves

There are limitations with Quadratic Bezier curves, for example we cannot trace a loop with a Quadratic Bezier curve.

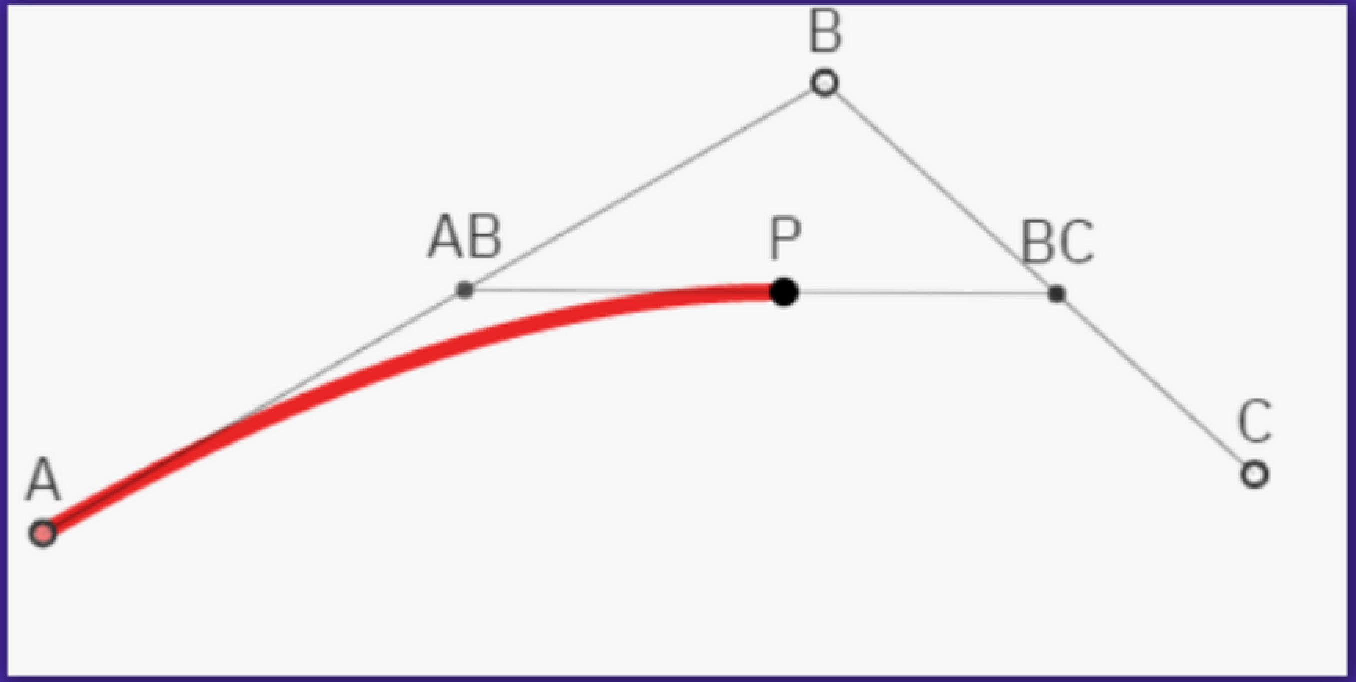
If we re-arrange Quadratic Bezier Eqn we get:

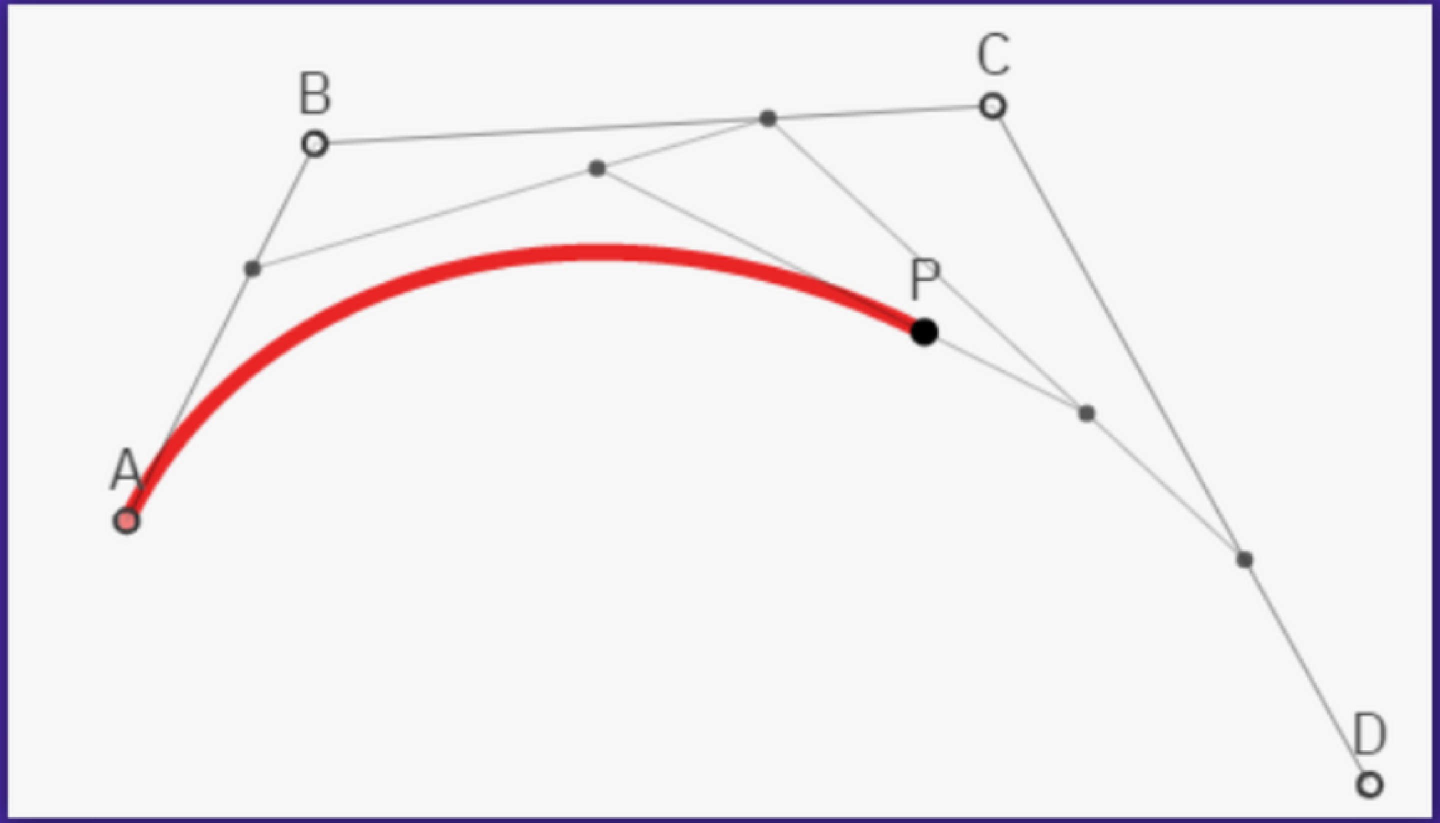
$$P = \{ A \times (1-t) + B \times t \} \times (1-t) + \{ B \times (1-t) + C \times t \} \times t$$

We can call the terms in { } to be new points:

$AB = A \times (1-t) + B \times t$

$BC = B \times (1-t) + C \times t$ $\Rightarrow$ **$P = AB \times (1-t) + BC \times t$**



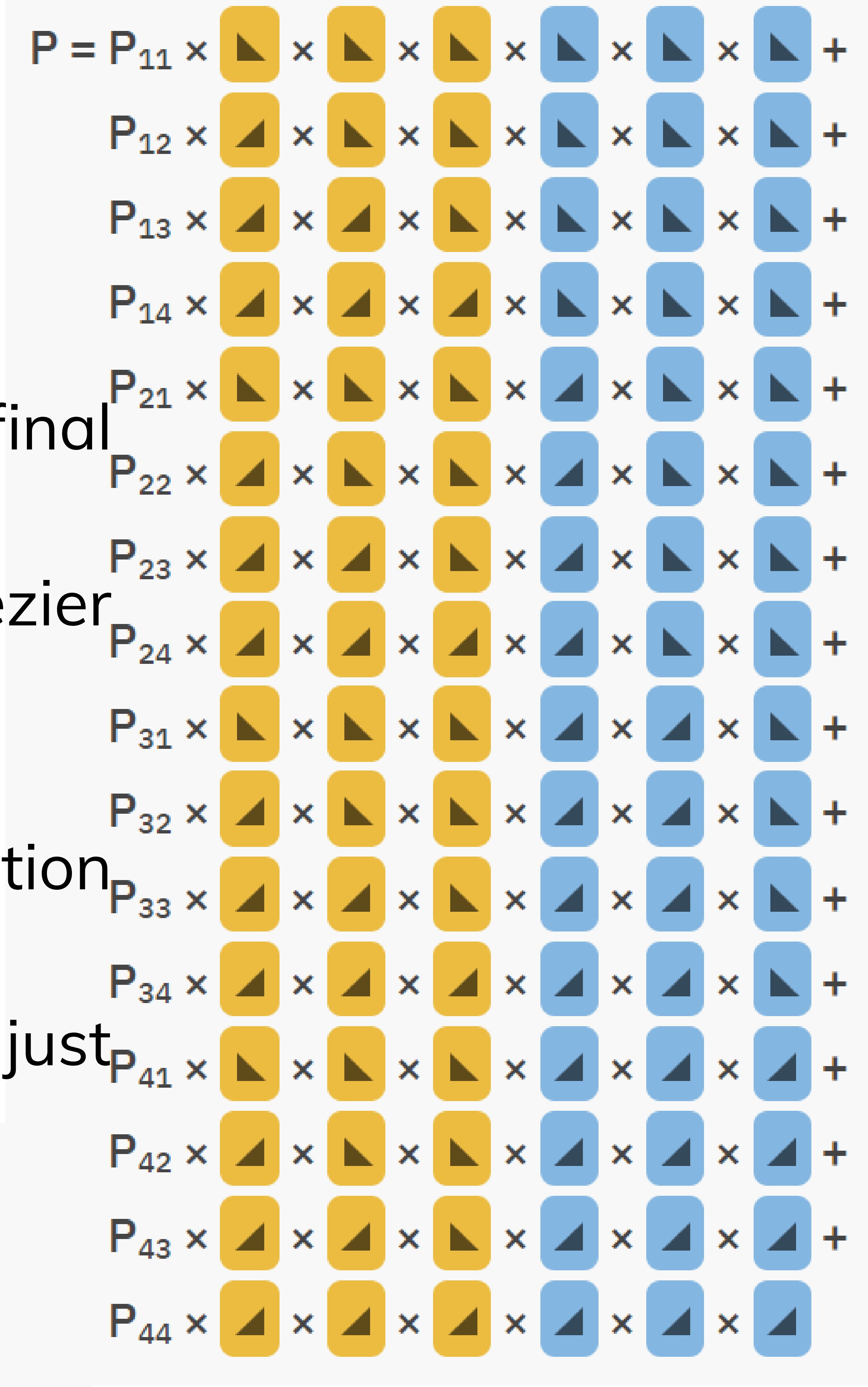


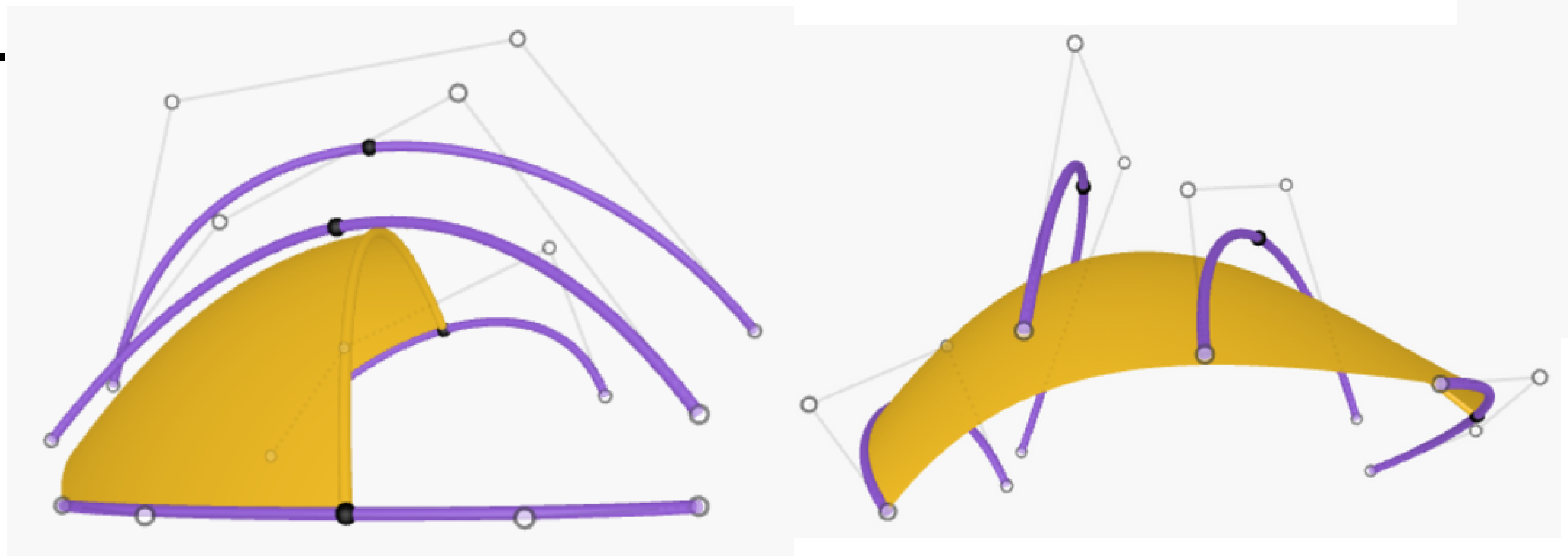
If we continue this interpolation we get:

$$P = \{ 1 \times A \times (1-t) \times (1-t) \times (1-t) \} + \{ 3 \times B \times (t) \times (1-t) \times (1-t) \} + \{ 3 \times C \times (t) \times (t) \times (1-t) \} + \{ 1 \times D \times (t) \times (t) \times (t) \}$$

Bezier Patches

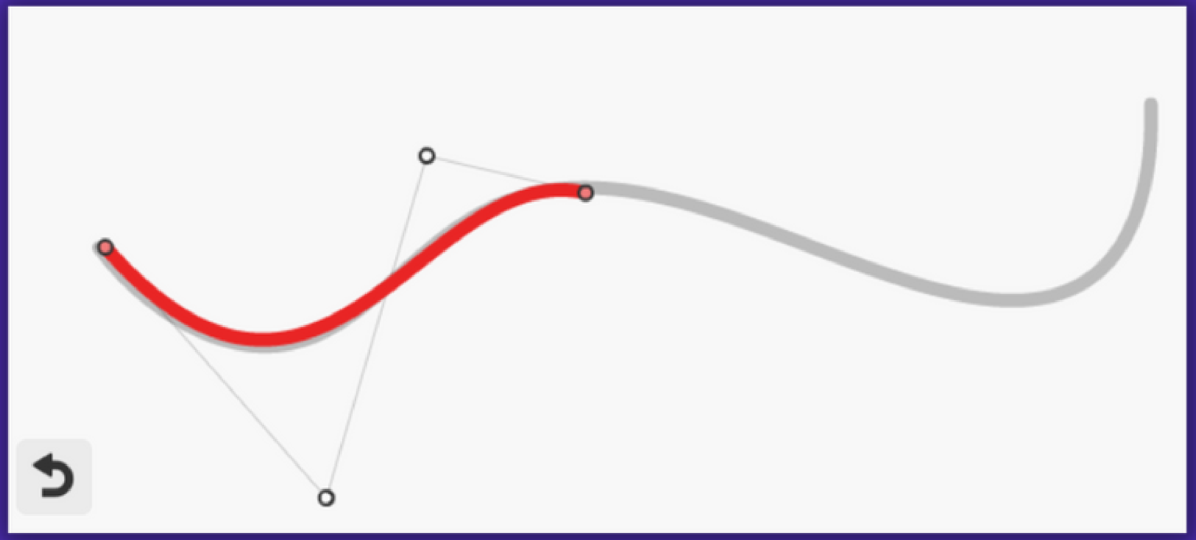
- Four Bezier curves in 3D space.
- These curves are used to rail **FOUR** control points.
- These control points in turn define the new Bezier curve and the final surface traverses this curve in space.
- Control points change their positions as they slide on four Bezier curves . **THIS forms the Bezier Patch.**
- The surface formed is defined by 16 control points.
- The eqn driving the surface consists weighting function multiplication with 16 sets of points.
- Majorly there were only three unique shapes and the rest were just rotations and reflections.



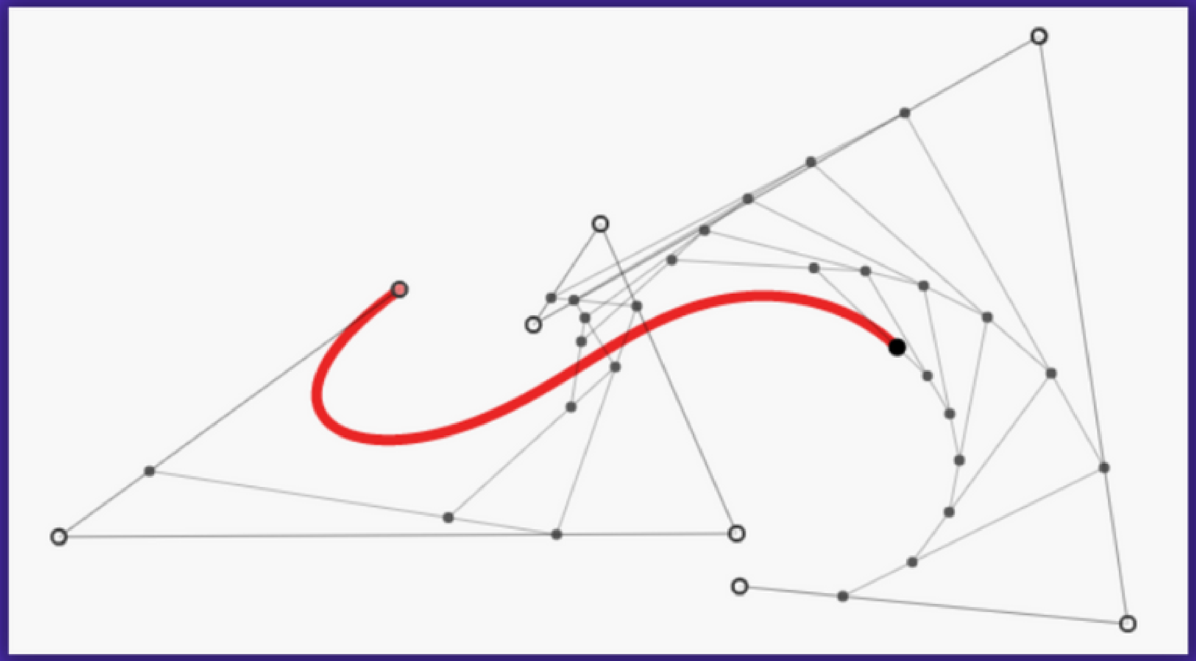


Beyond Cubic

Bezier(Cubic) curves can have at most two bends. As you can see in the figure alongside, we cannot retrace the S with the Bezier curve (in Red).



The above stated problem can be resolved by creating more control points, however Bezier curves with more control points provide better flexibility but they have their own problems.

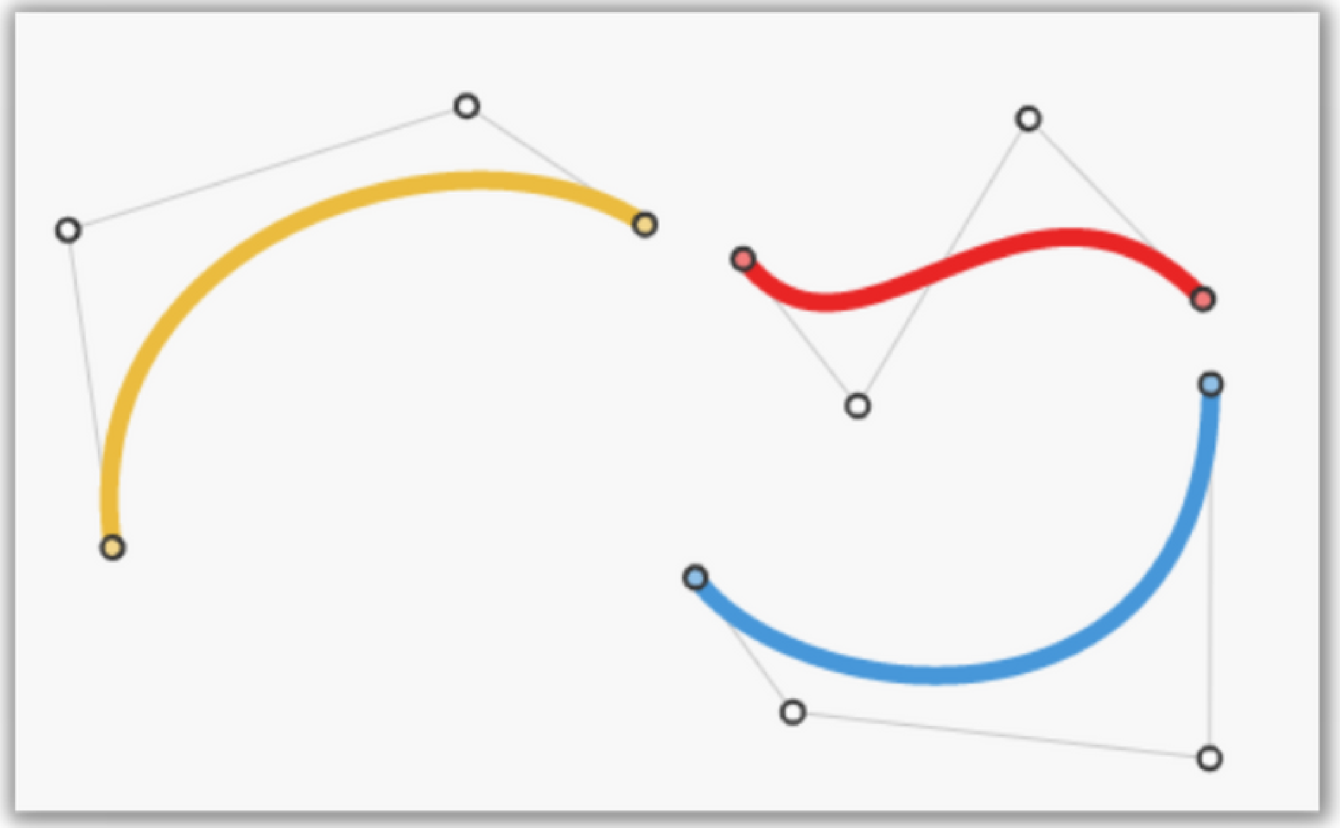


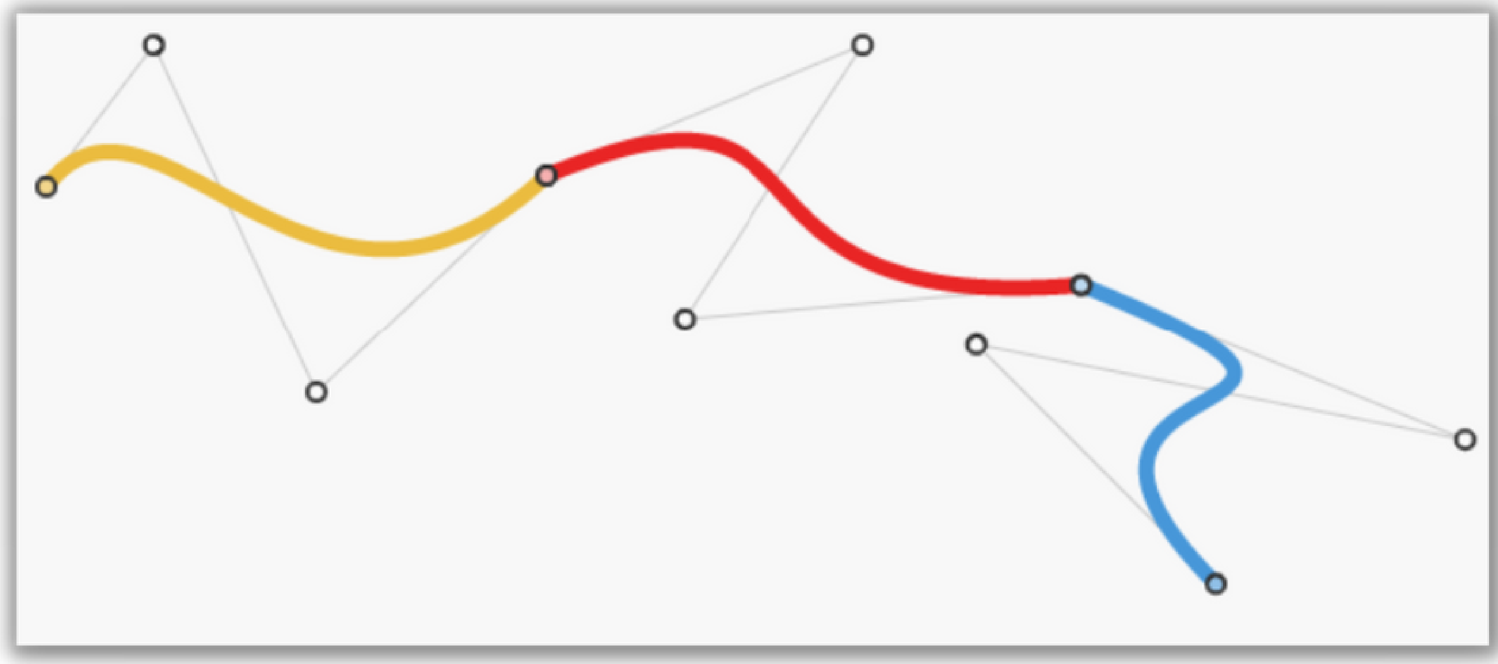
One of such problems is that a Bezier curve with many control points lack "Local Control".

In a Bezier Curve with many control points, as we progress through the curve we see, to some extent they (each of control point) contribute almost the entire curve.

Complex Shapes with Bezier Curves

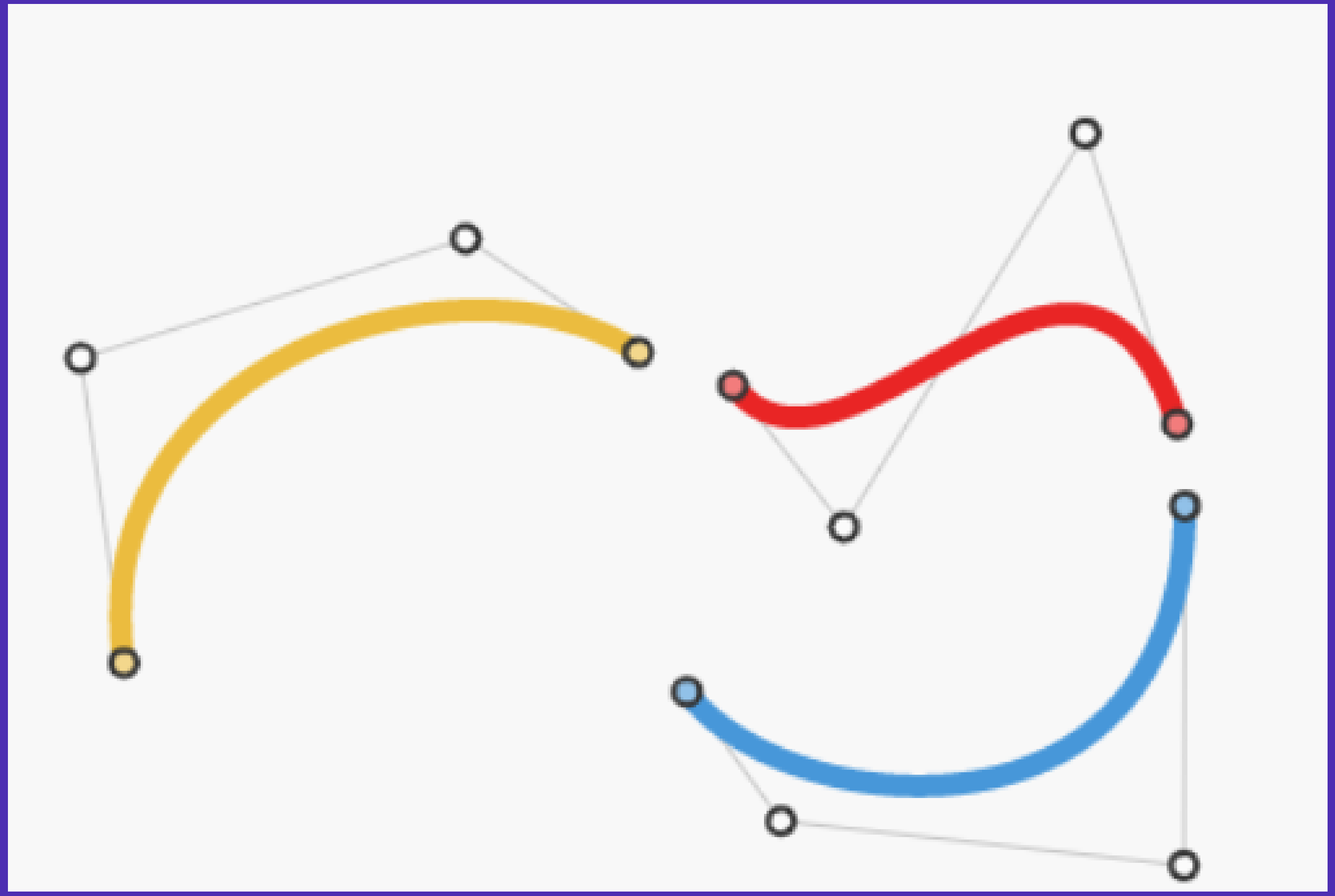
- The idea is to use more Bezier curves with fewer control points to achieve more flexibility and complex shapes.
- The problem with more curves is it becomes a little cumbersome and there are gaps as shown in the image alongside.
- The above stated problem can be resolved by joining the end points of the curves.
- Point 3 solves the issue of gaps however another issue arises from this solution, what we now have are some sharp corners which we need to account for (if we aim for smooth nice curves).
- To ensure smoothness we need to ensure that 'tangency' of the neighboring curves is continuous.

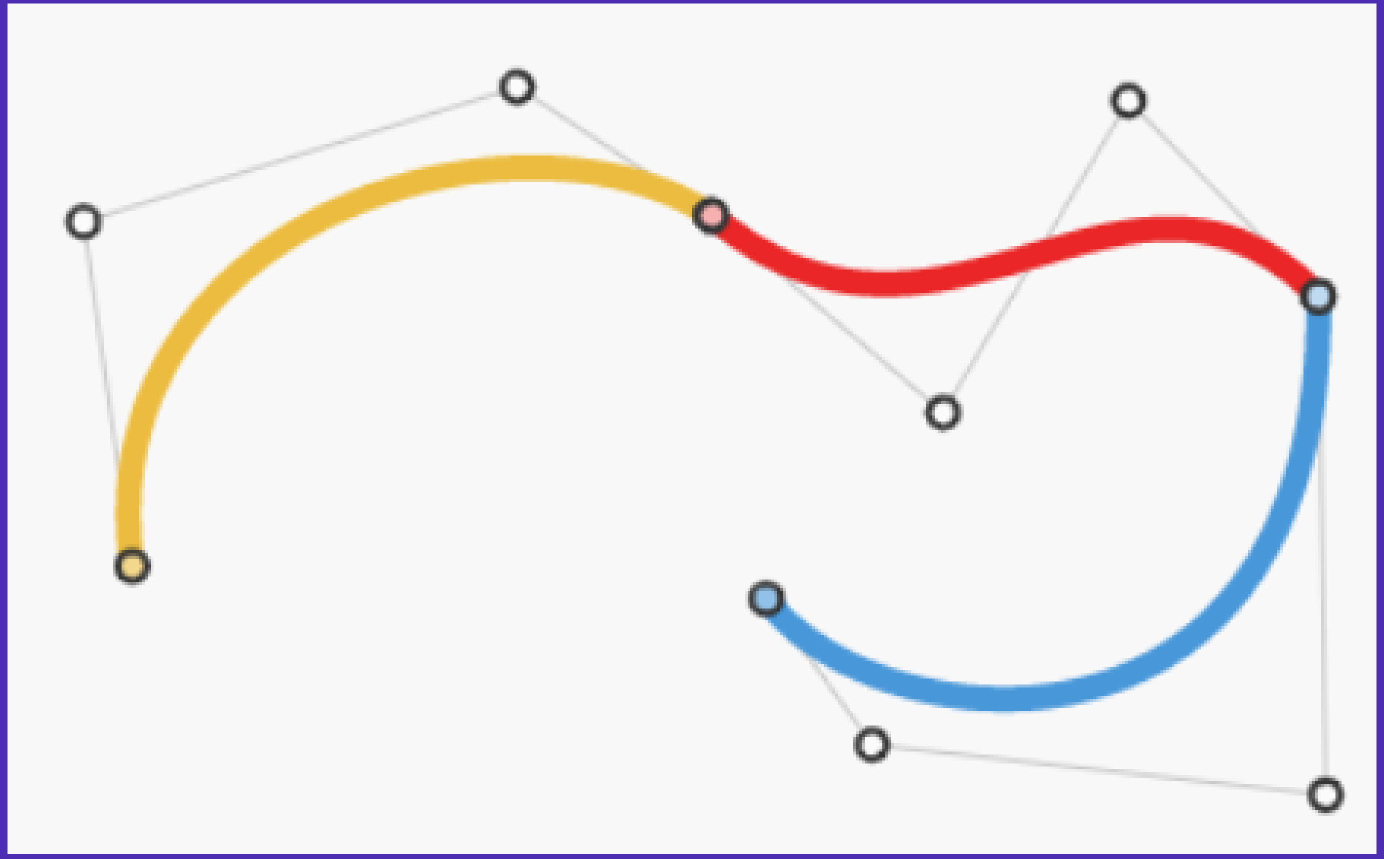




Splines

- Instead of using a single curve with many control points, we can use multiple curves with few control points.
- While flexible, the setup of three cubic Bezier curves is a bit cumbersome to use, as these three curves have gaps unless we connect their end points.
- This eliminates the gap problem, but if we want good, smooth forms, we'll still need to account for certain sharp edges. We must verify that the "tangency" of surrounding curves is also continuous in order to eliminate the kinks.

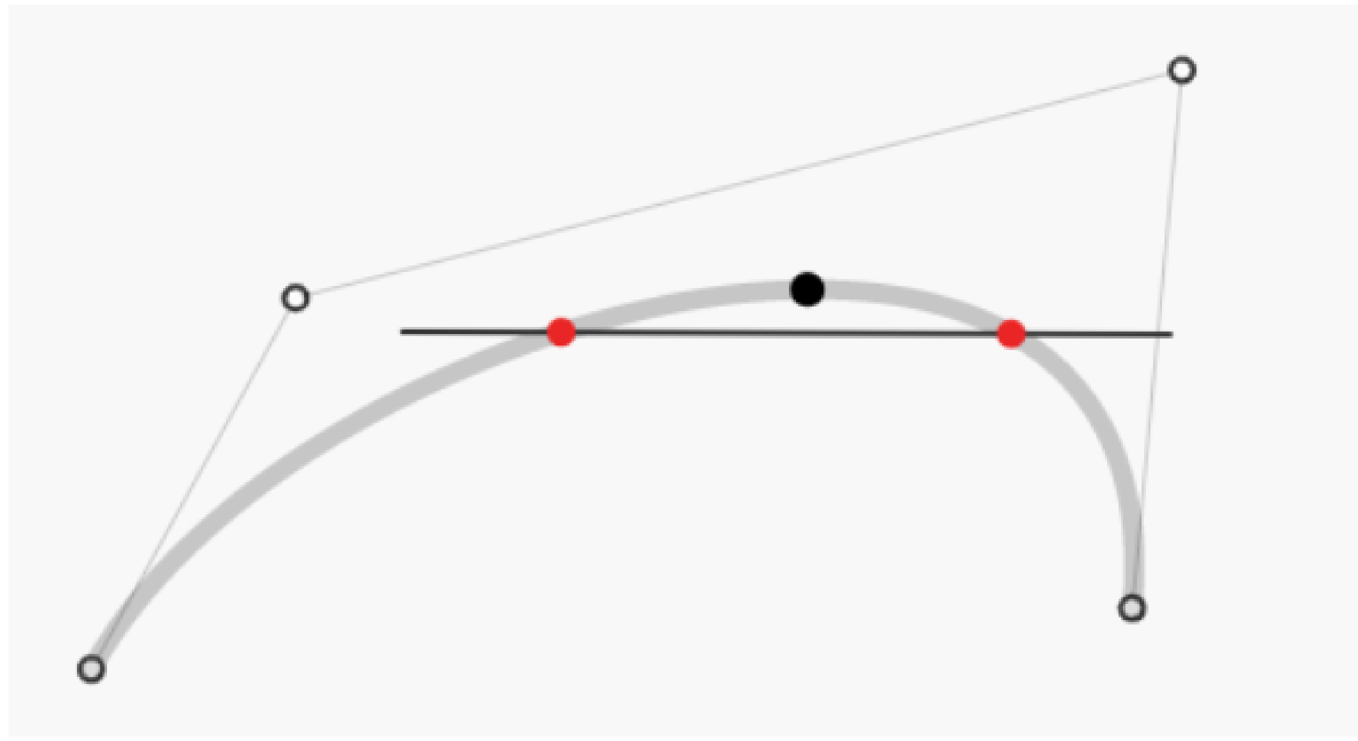


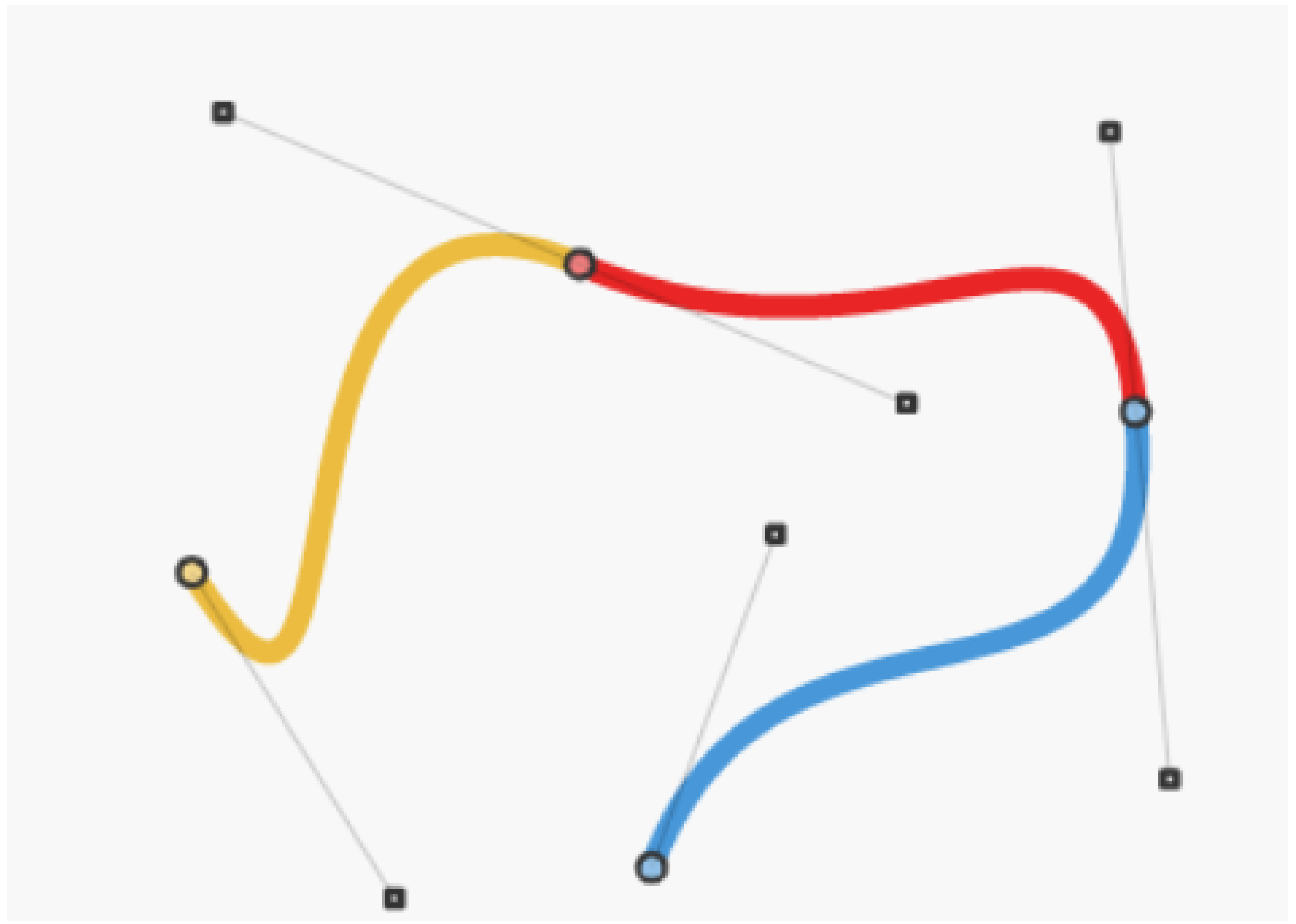


Splines (Contd.)

Tangency

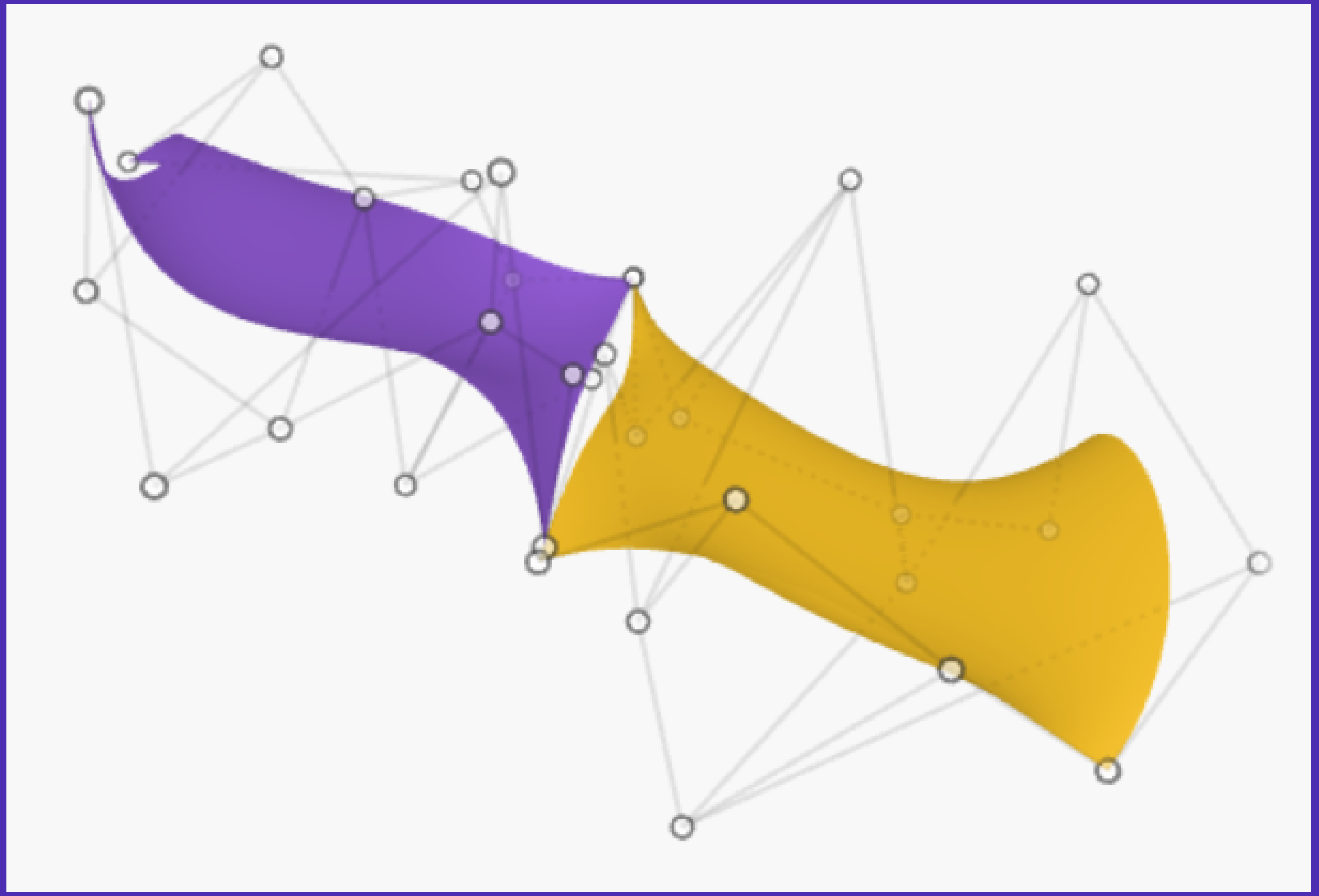
- A tangent is a local direction that is "straight forward." It's easy to find simply moving two beads around a curve. The line spanning their centres approaches the tangent direction as the beads move closer.
- The next and previous neighbours of the connecting control point must be on the same straight line to guarantee that the tangency of nearby cubic Bézier curves is constant.
- Many design tools connect the control points of adjacent curves to form a single control axis that keeps the tangency.

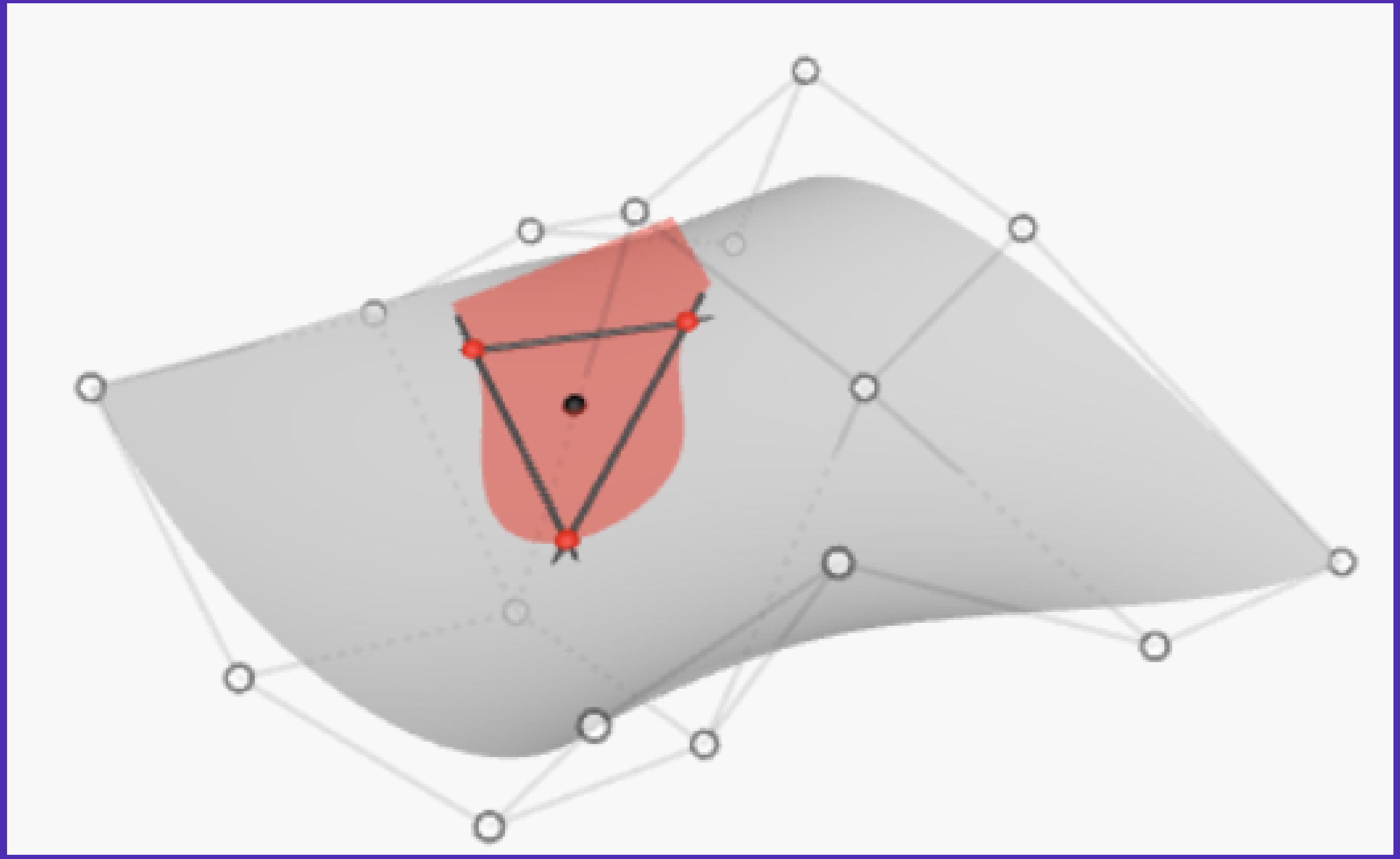




Splines (Contd.)

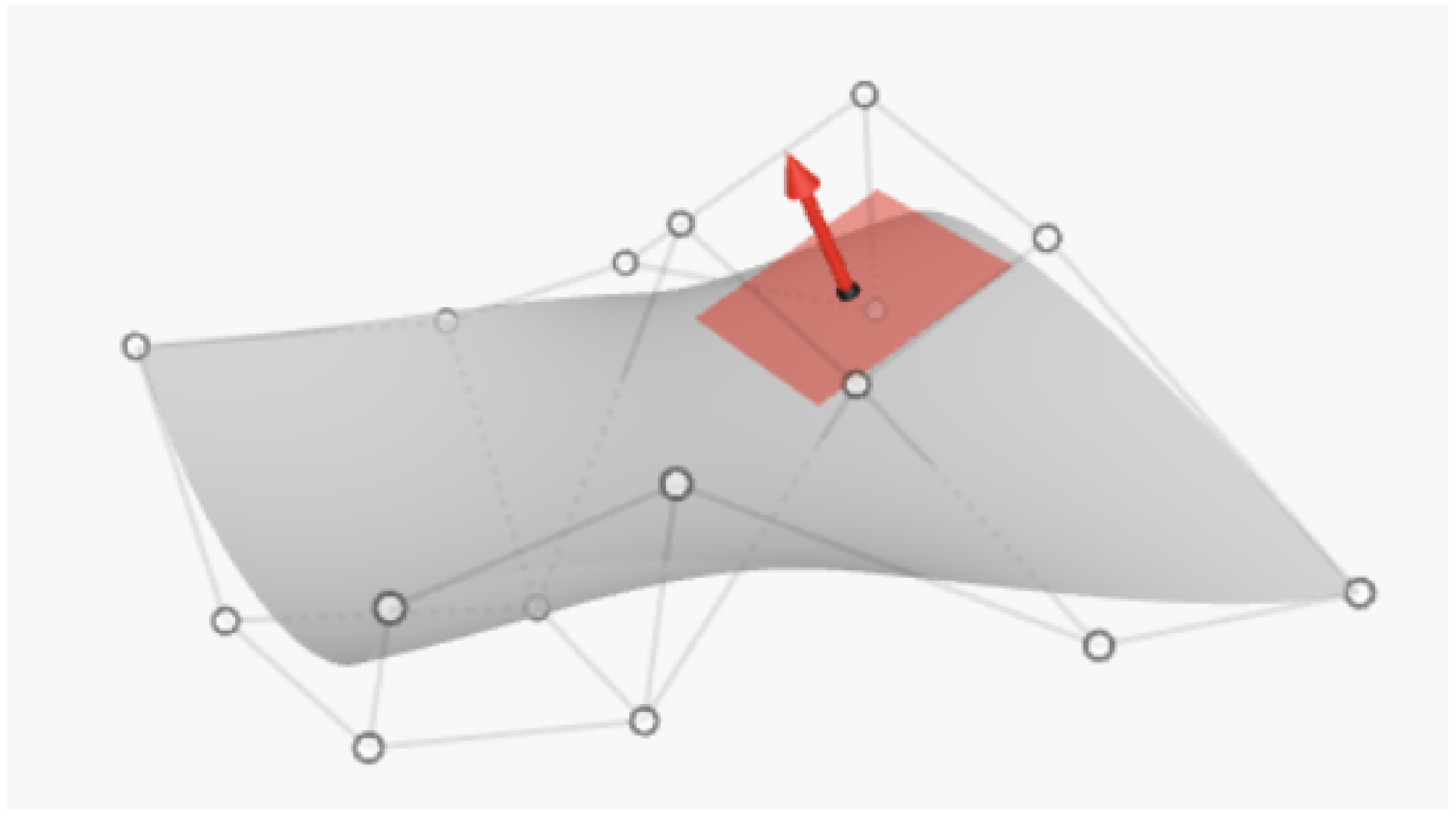
- Splines are piecewise polynomial curves that are made up of many components, each of which is characterised by a polynomial function.
- When it comes to designing complex surfaces, we can employ the same idea of using multiple cubic Bézier patches to form more complicated shapes.
- When it comes to tangency, the idea of a single tangent direction for surfaces doesn't make much sense because the surface "goes" in multiple directions at any one place.
- We define a full tangent plane to account for this.
- We may take three beads and three straight wires linked to them and slide them near to the point on the surface, similar to how we did with curves.

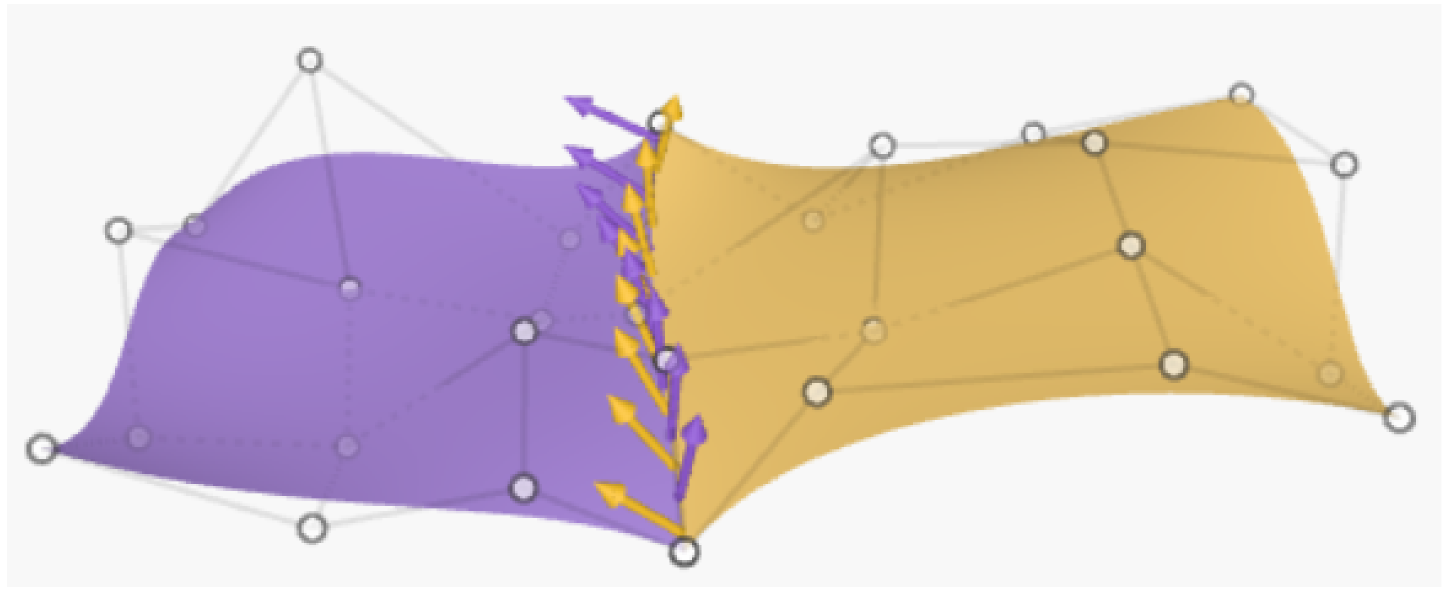




Splines (Contd.)

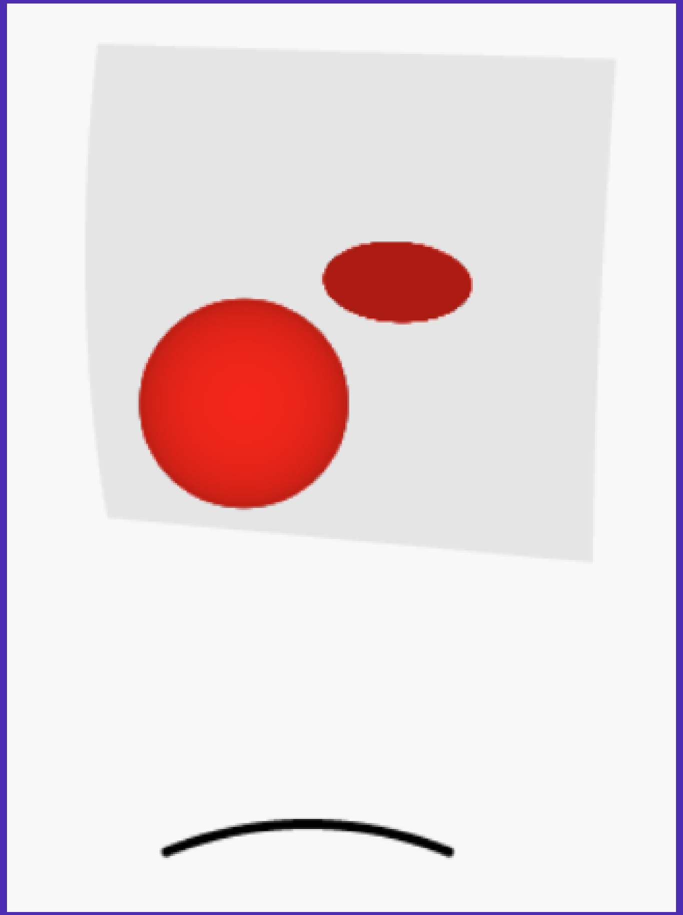
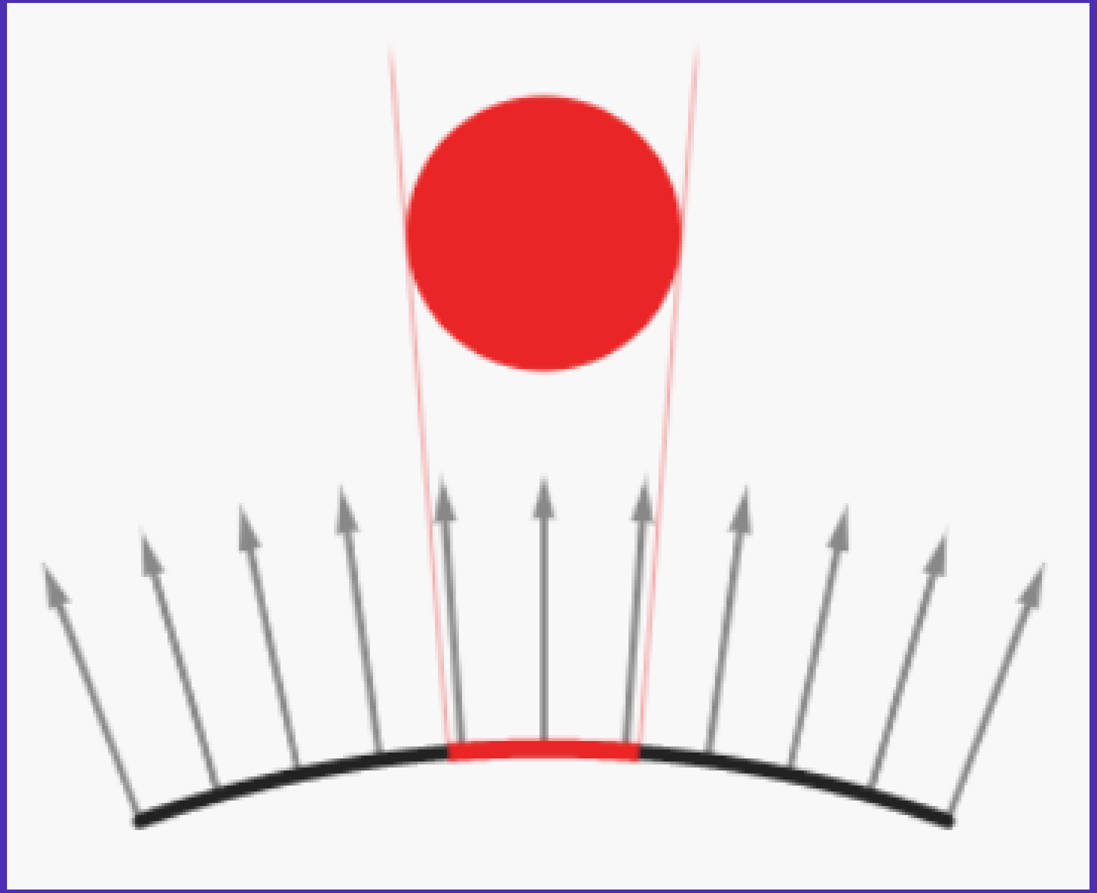
- The tangent plane is defined by the three wires as the beads approach the goal point infinitely near. That plane also has a perpendicular direction, which we refer to as the surface's normal direction at that point — I've shown that with a red arrow.
- The normals at the joining edge must be oriented in the same direction for two surfaces to be linked without a sharp edge. Note that even if the surfaces do not have any gaps, it takes a lot of modifications to maintain them connected and free of sharp edges.
- But it isn't the end of the problems that coupled patches can cause.





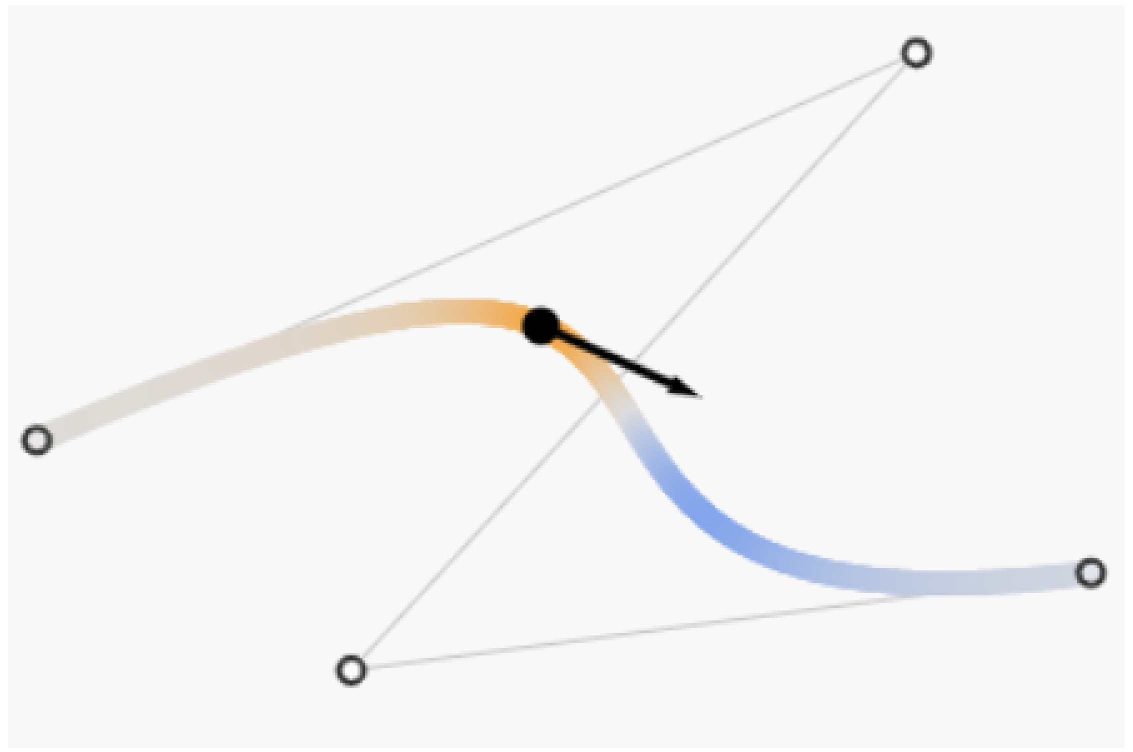
Curved Mirror on The Wall

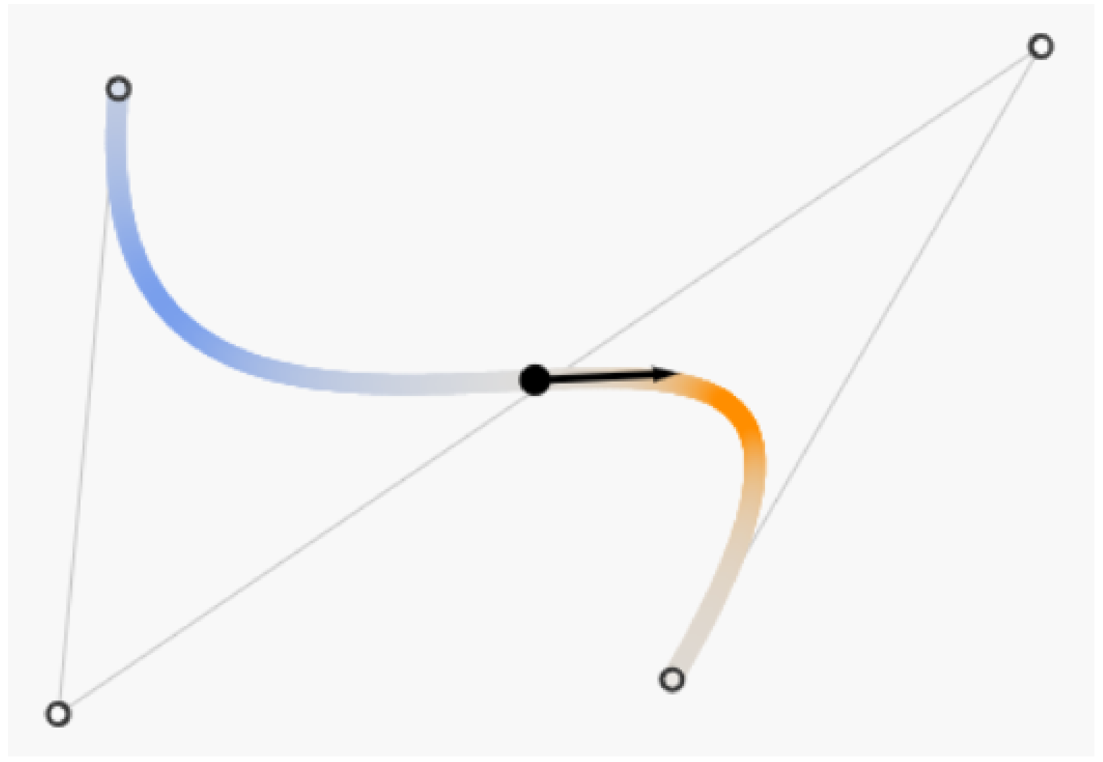
- It would be logical to believe that removing sharp edges is all that is required to make attractive surfaces, but we still have work to do when the surfaces are reflecting. A red sphere is reflected on a mirror-polished surface in the demonstration below.
- Notice how the sphere's reflection narrows as the mirror moves away from the flat form.
- We must examine the scenario from the side to have an intuitive grasp of why this occurs.
- The black line symbolises the mirror, while the red line depicts the sphere's reflection in that mirror.

Curved Mirror on The Wall (Contd.)

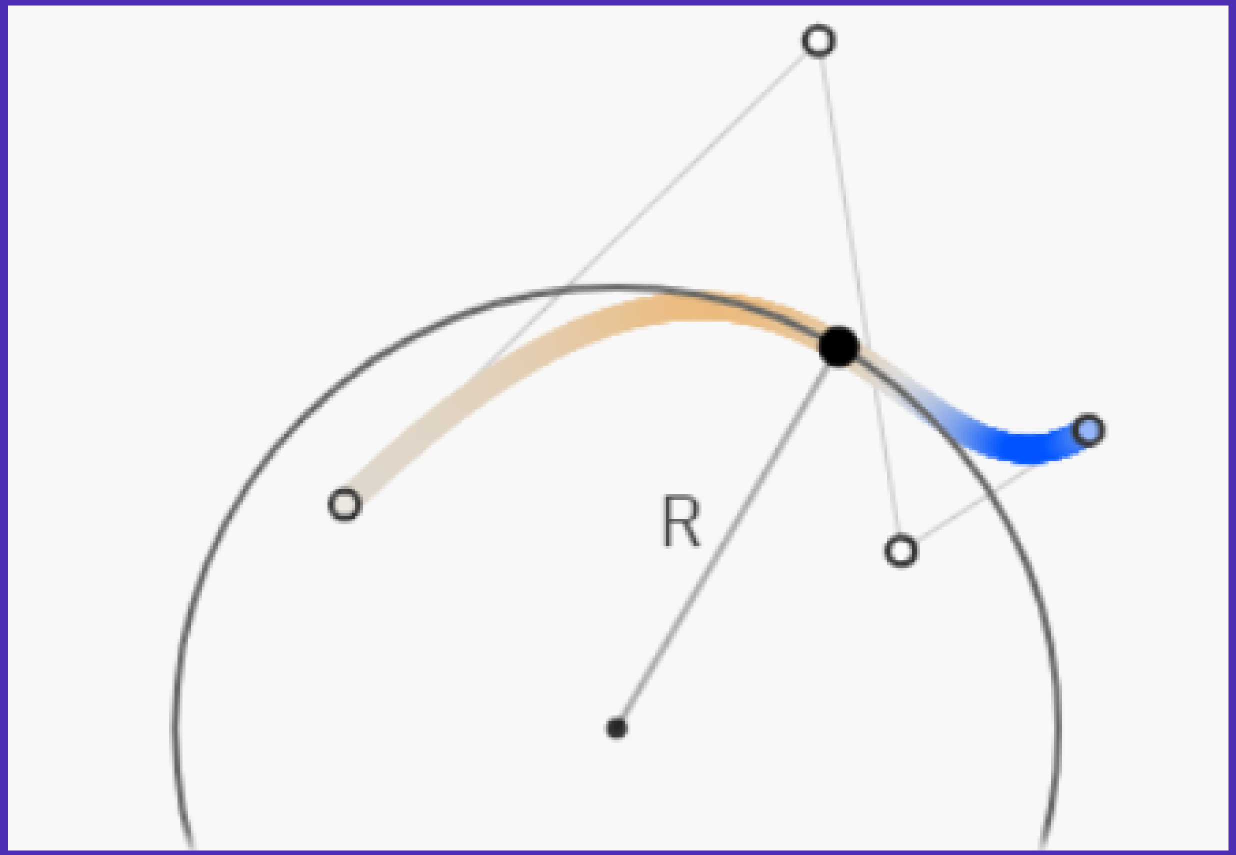
- The normal directions of the surface spread out as it bends.
- In basic terms, a curved mirror "sees" more of its surroundings, but because the mirror's area remains constant, the sphere's reflection must shrink to fit in.
- When we bend a surface, we're actually modifying the profile of the surface's curvature. Curvature is a term that describes how rapidly a curve changes direction.
- I've coloured in curving areas of the curve in the demonstration below; the higher the curvature, the more vivid the colour. Because this curve may bend in two distinct directions, I've used the colours orange and blue to differentiate between them.
- You can see how rapidly the local orientation of the curve, indicated as a black arrow, changes in places with considerable curvature by sliding the slider:

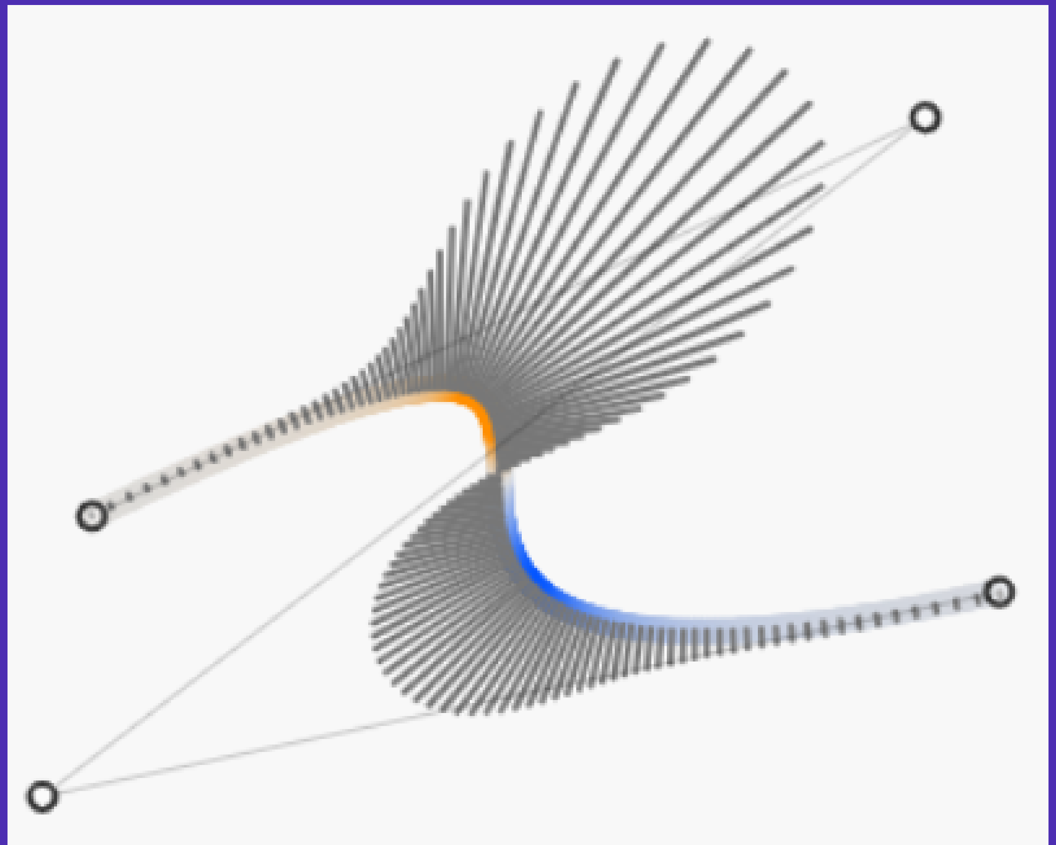




Curved Mirror on The Wall (Contd.)

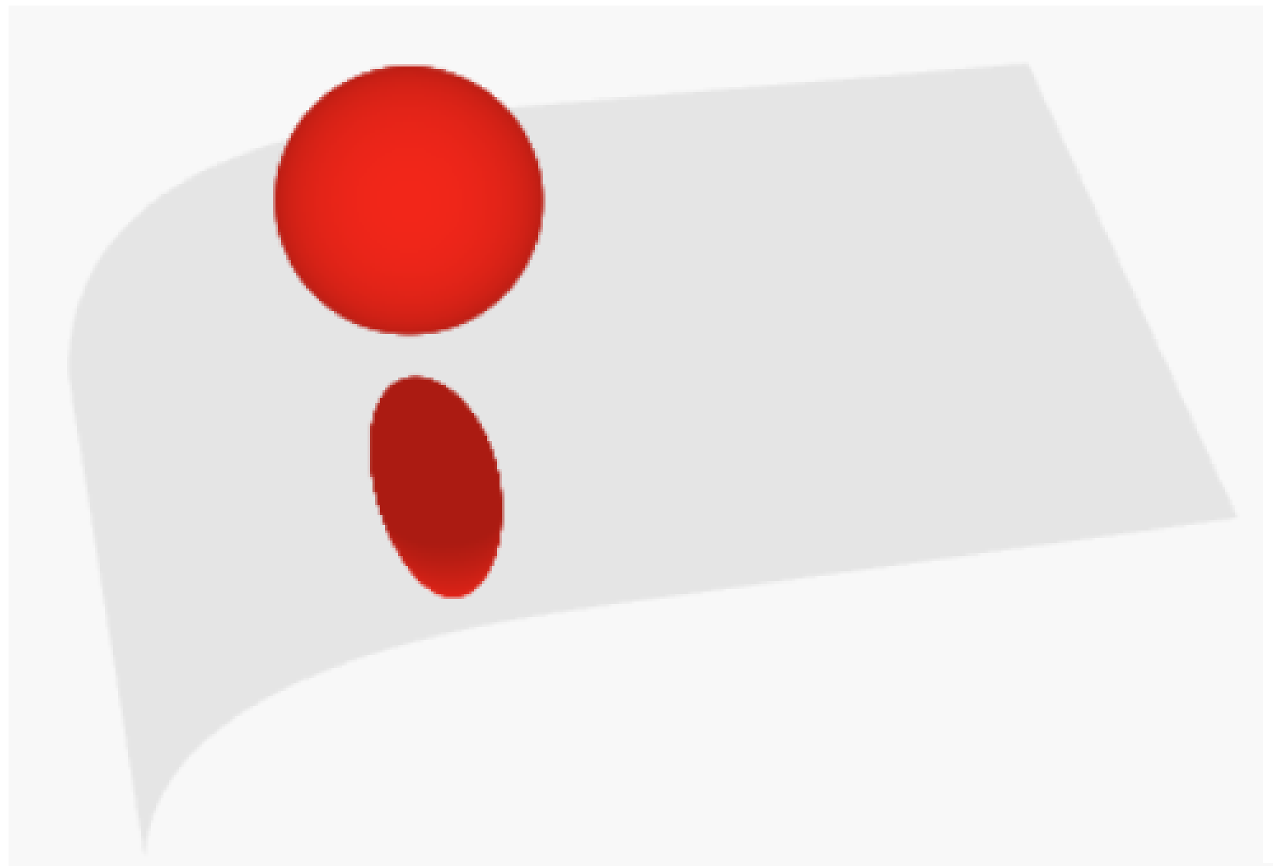
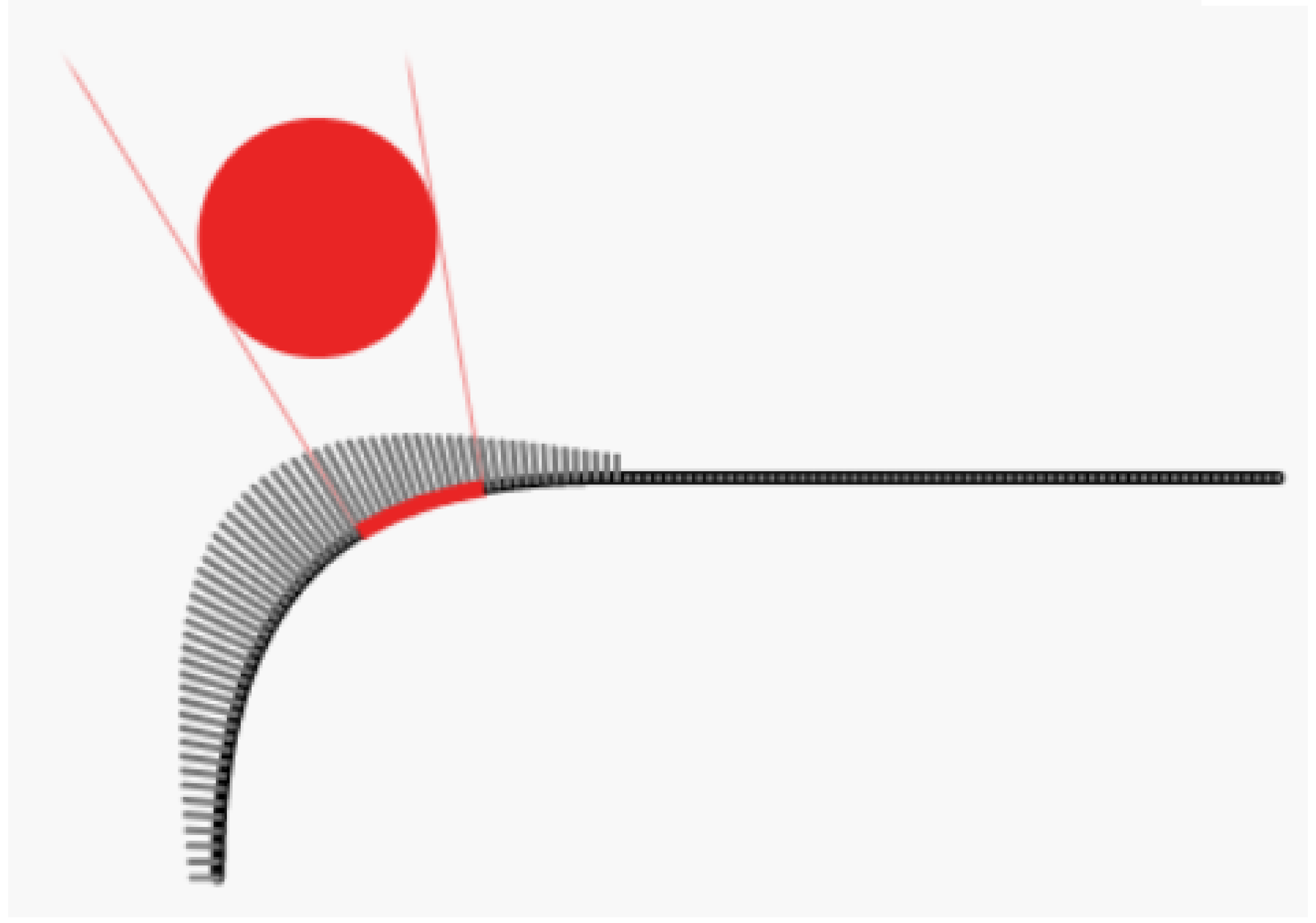
- A straight line has no curvature, and the arrow never turns, but a tight curve has a significant curvature, and the arrow rotates extremely quickly as it passes through it.
- An osculating circle, which is a circle that "fits" into a curve locally, can be used to illustrate a more exact assessment of a curvature.
- Curvature κ is the inverse of the radius R of that circle, that radius R is also known as the radius of curvature.
- It's worth noting that the inverse relationship makes sense: a small curvature predicts a curved region with a big radius of curvature. A curvature comb is also used to visualise the curvature of a curve.





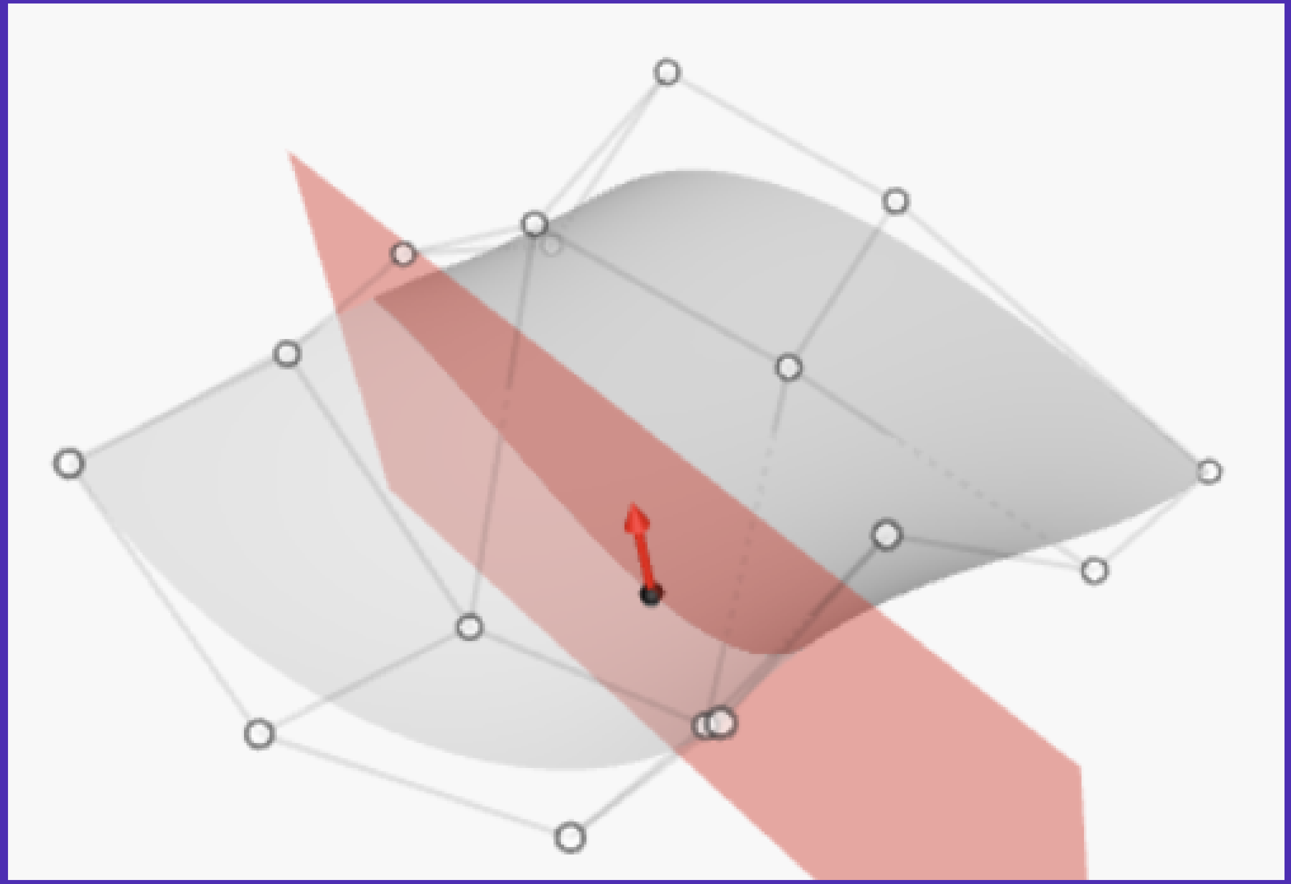
Curved Mirror on The Wall (Contd.)

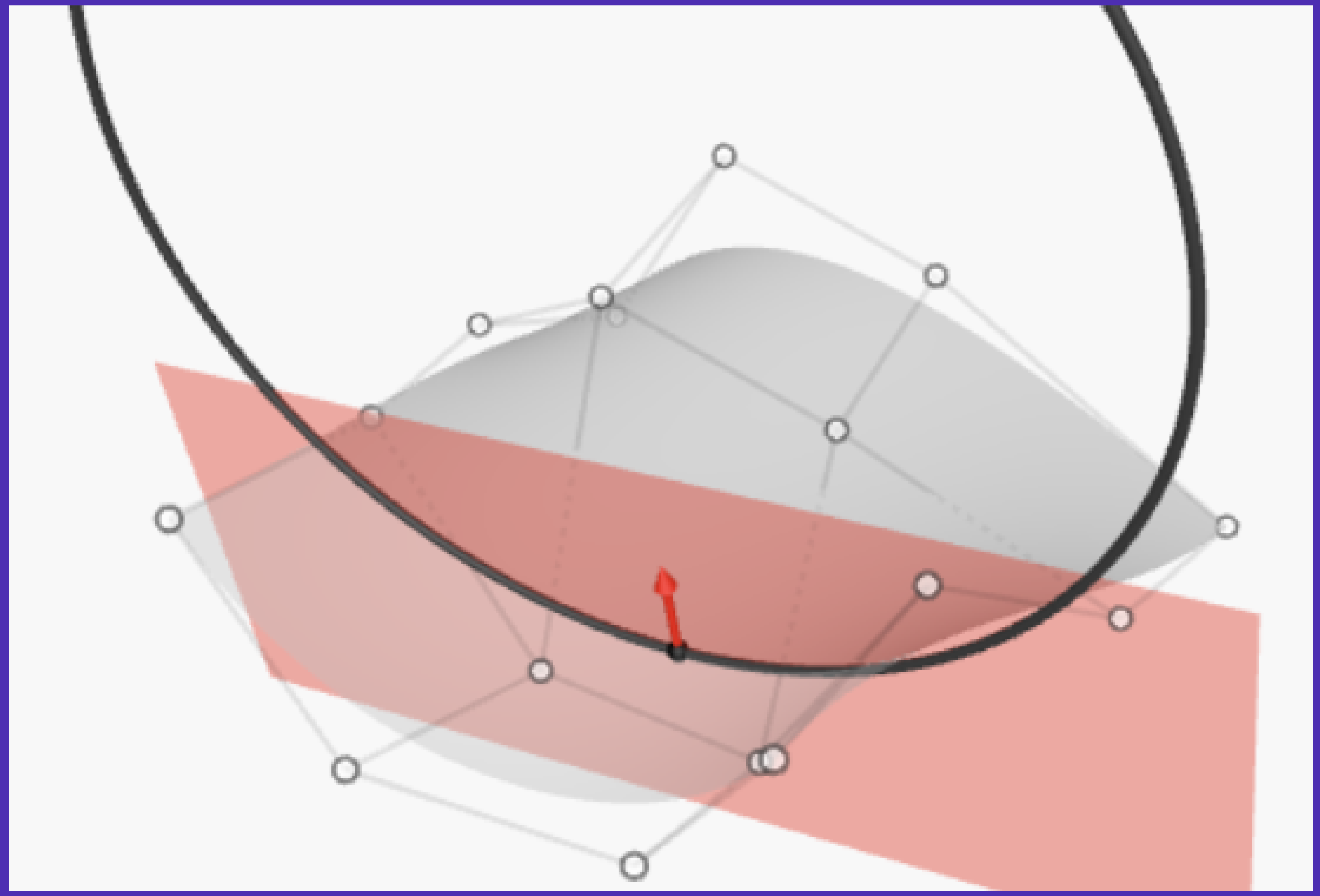
- Comb's teeth protrude in the curve's usual direction, and the longer a tooth is, the greater the curvature in that location.
- The top slider determines the sphere's location, while the bottom slider determines how closely the two surfaces' curvatures match.
- It's worth noting that the connection has no sharp edges.
- The reflection of the sphere at the connection might seem bizarrely twisted into a UFO-like shape for varied values of the curvature match, or it can be relatively smooth.
- We can obtain a better sense of what's going on if we look at the issue from the side with a curvature comb visible.

Curved Mirror on The Wall (Contd.)

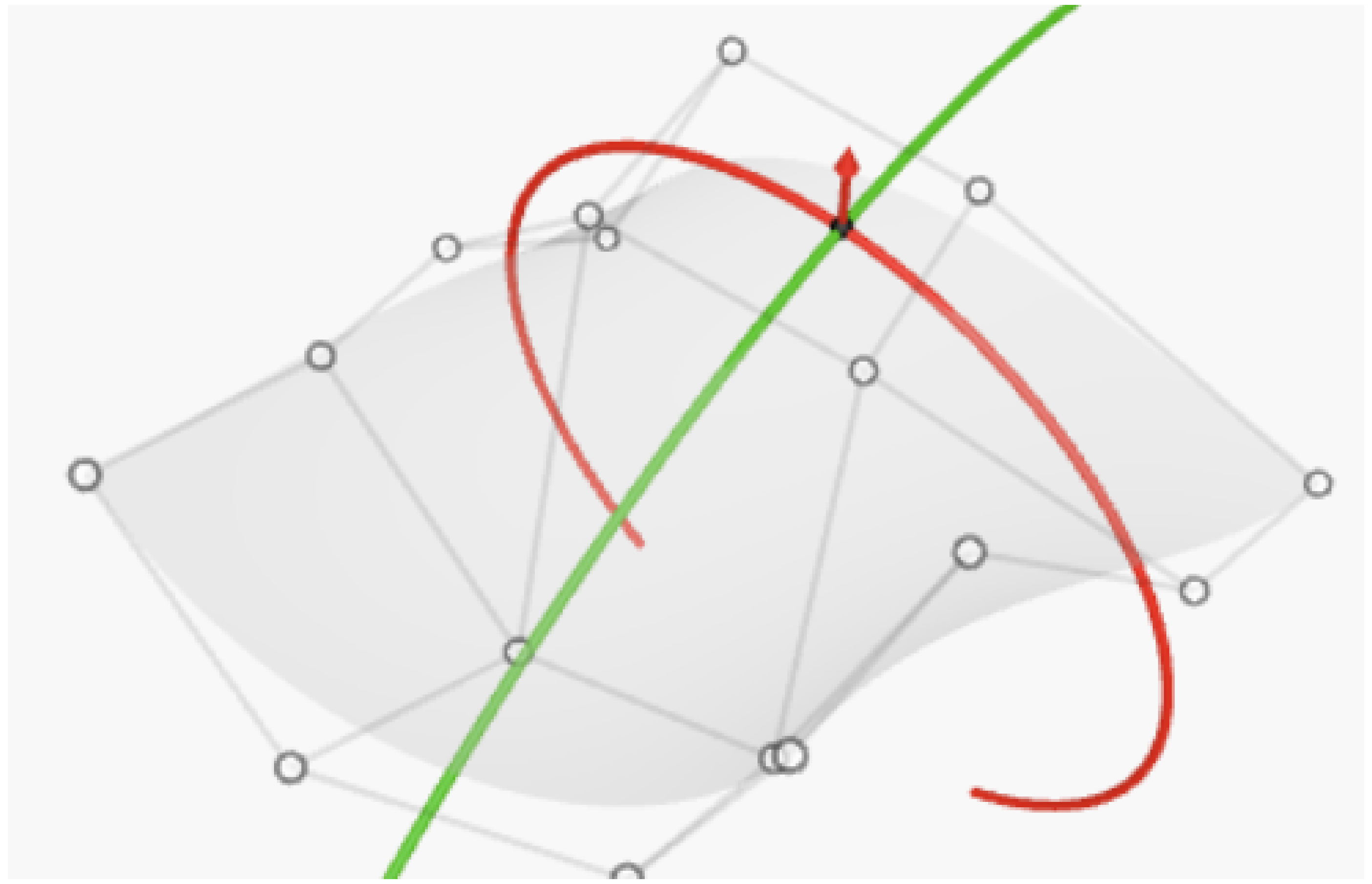
- For every point on a surface we can take a normal direction and use it as pivot on which a new plane can rotate.
- Notice that in some orientations you can see a fairly sharp falloff in the shading on the surface. That transition gets much smoother if we make the curvatures vary more smoothly.
- Even though we were looking at light reflected off surfaces, we've actually been talking about the curvature of the curve defining the profile of those surfaces.
- Notice that this plane slices the surface creating a curve at the cross section. We can then find the curvature of that curve at that point and show the osculating circle, or at least a part of it

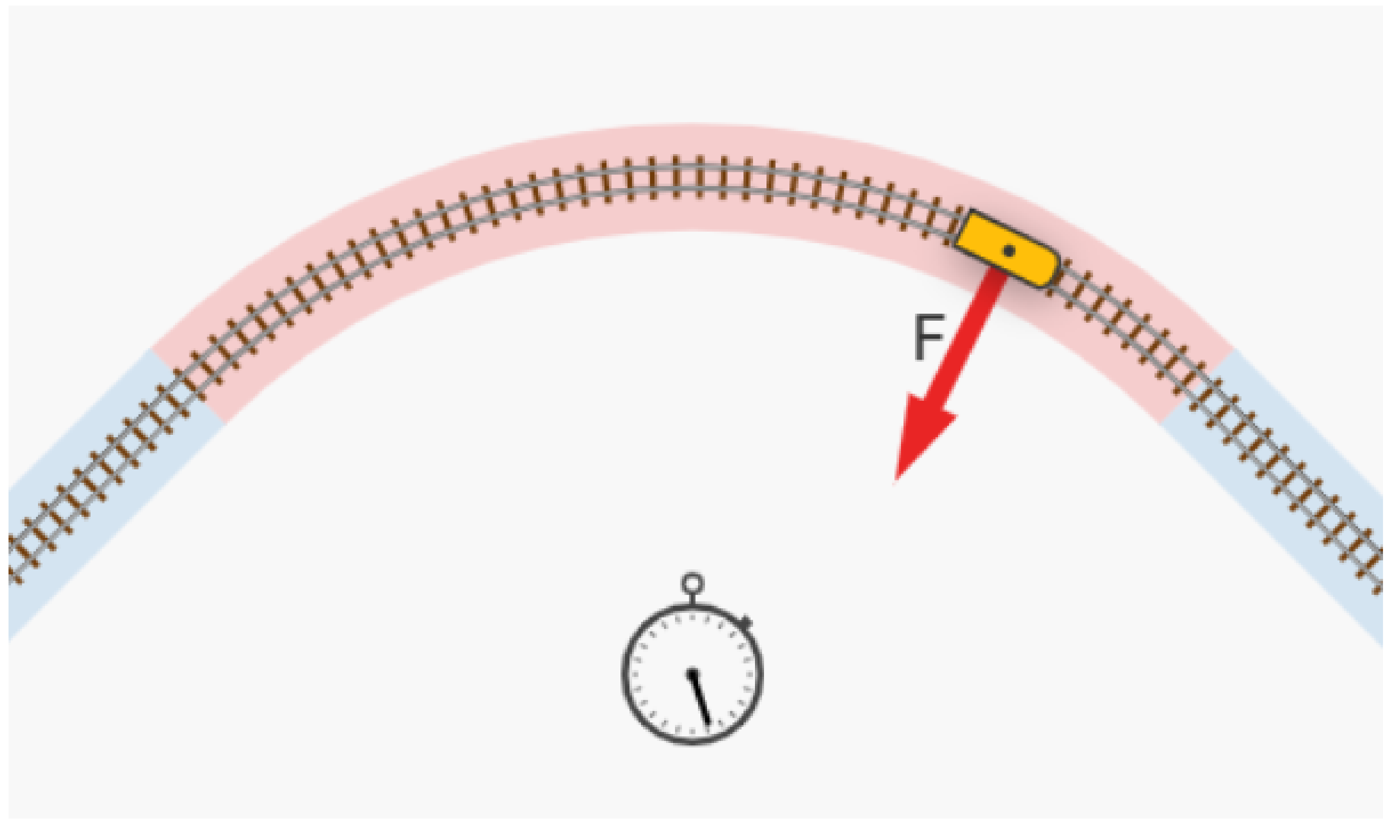




Curved Mirror on The Wall (Contd.)

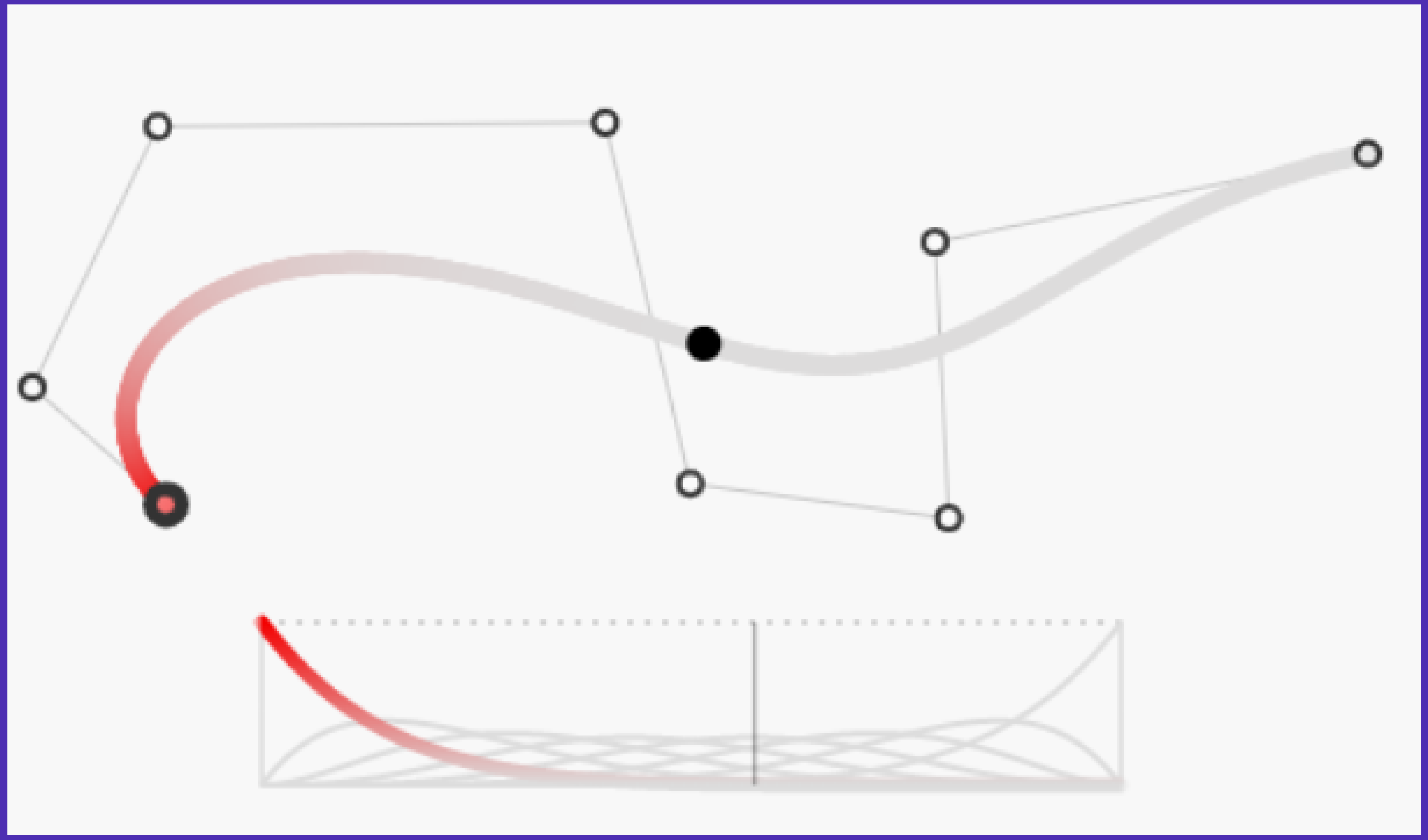
- For some cylindrical shapes, one of the curvatures will always be 0, but for saddle-like surfaces one will be positive and one negative. The directions of two principal curvatures are always perpendicular to each other, but the reason for that is a bit complicated. In the demonstration below I put a little train riding on that track, and you can control the time using the slider.
- A centripetal force causes the train and its passengers to reverse direction, as seen by the red arrow. The object's mass m, its velocity V squared, and the radius of curvature R of the travelled route determine the force F.

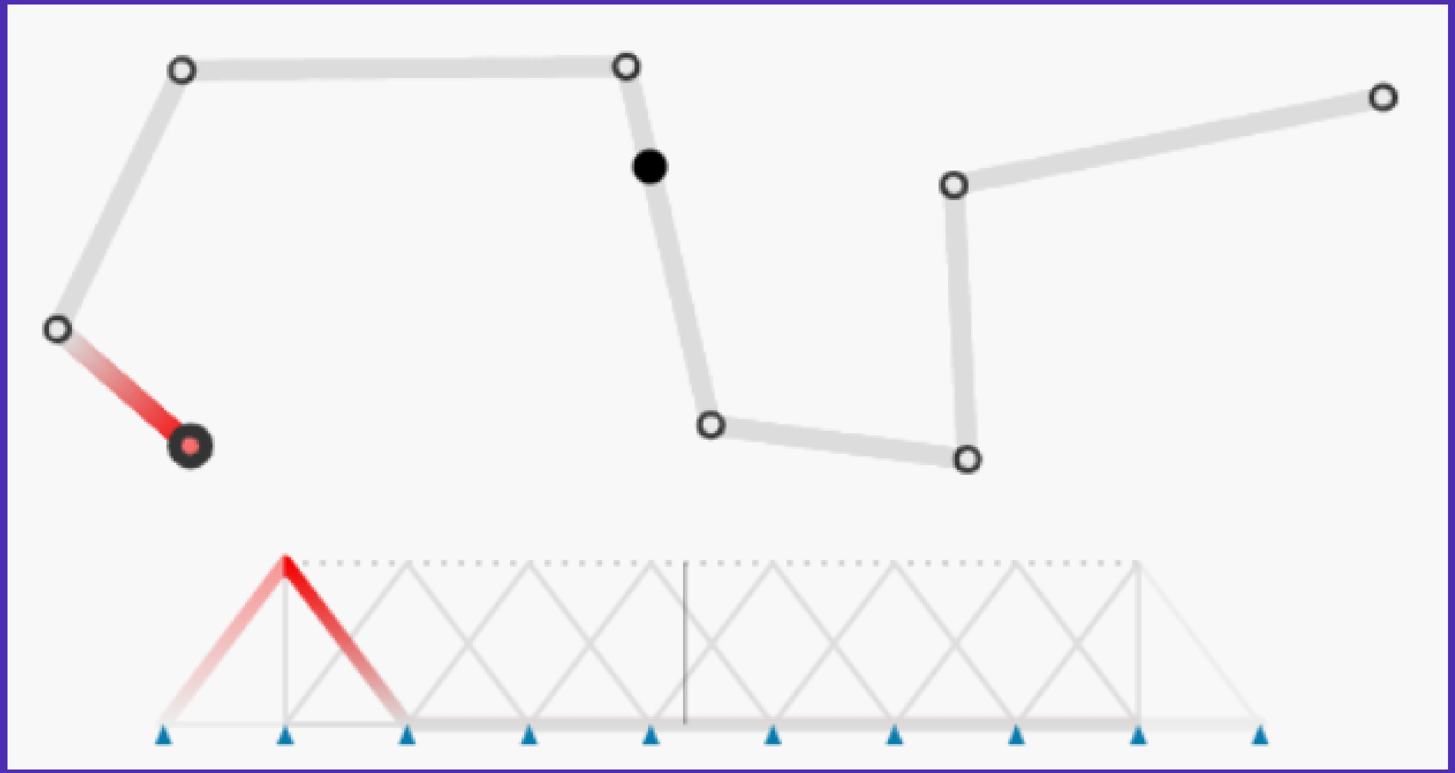




Building The Basis

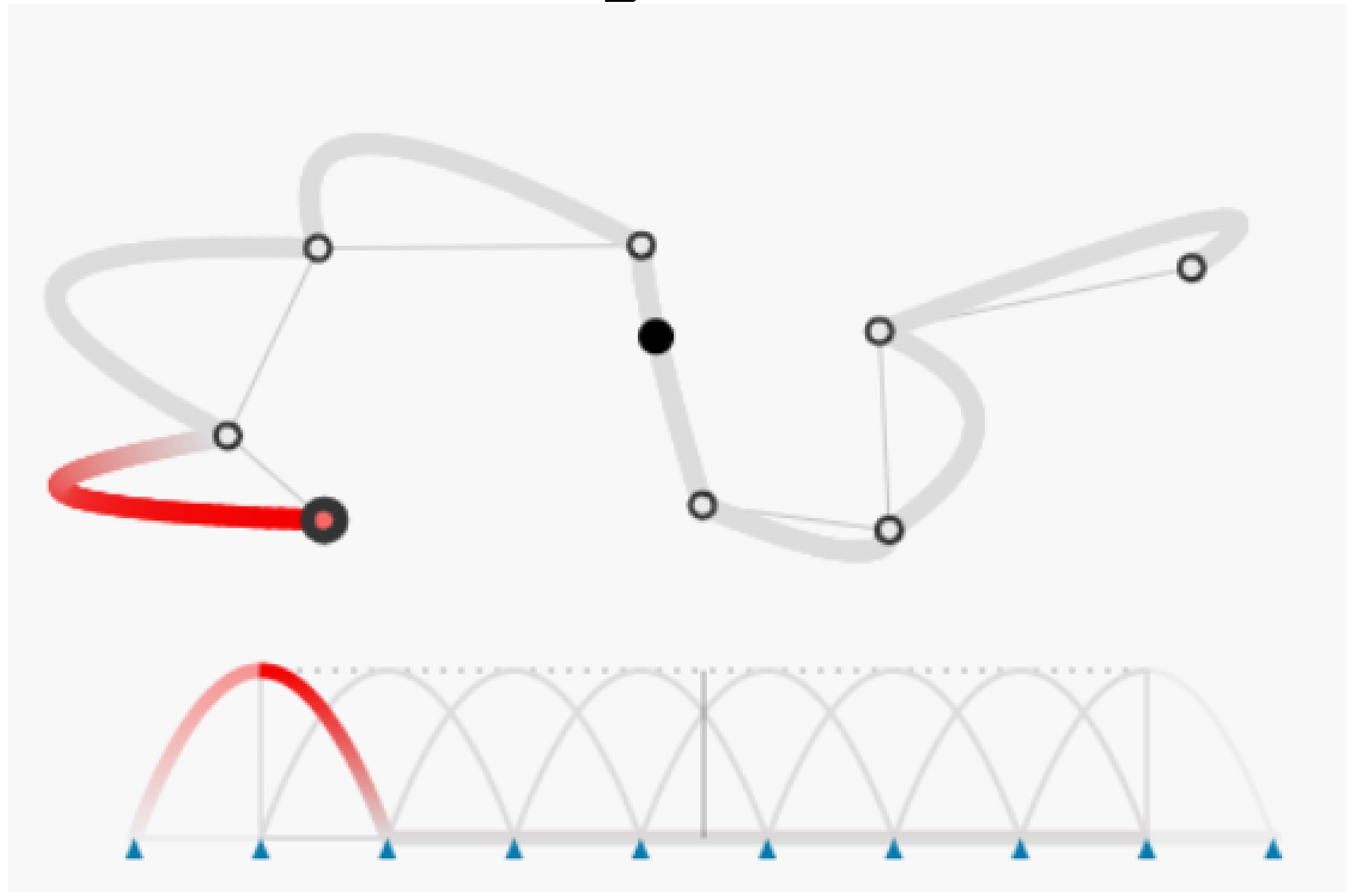
- The weight of a control point is non-zero only in a small range of the entire progress of the curve – each point has a minimized reach.
- We've not only created some weird arcs, but also their shape and "sidedness" depends on where on screen the control points are.
- One could be tempted to use an arbitrary shape for the individual weighting functions, for example, some smoother parabolas.

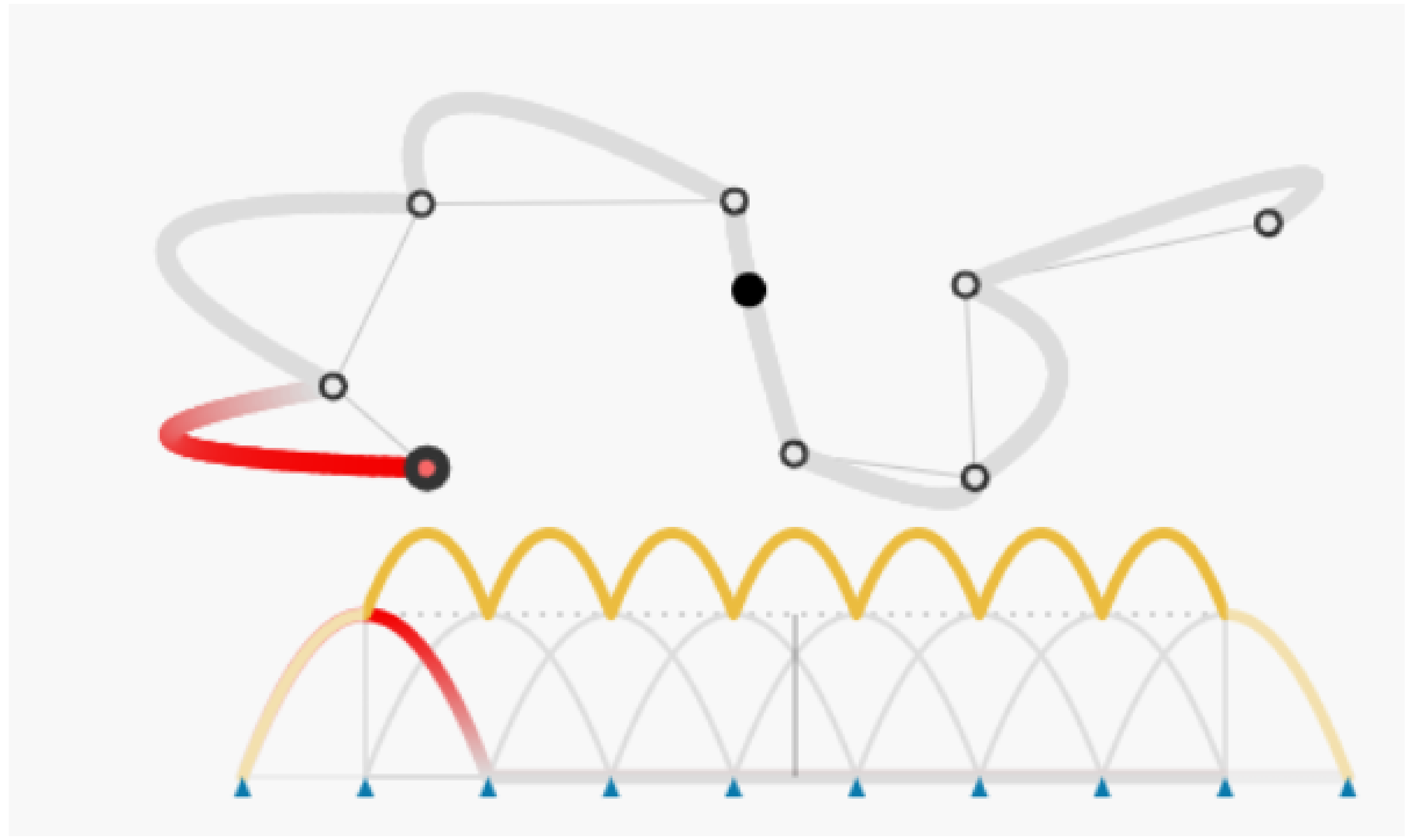




Building The Basis (Contd.)

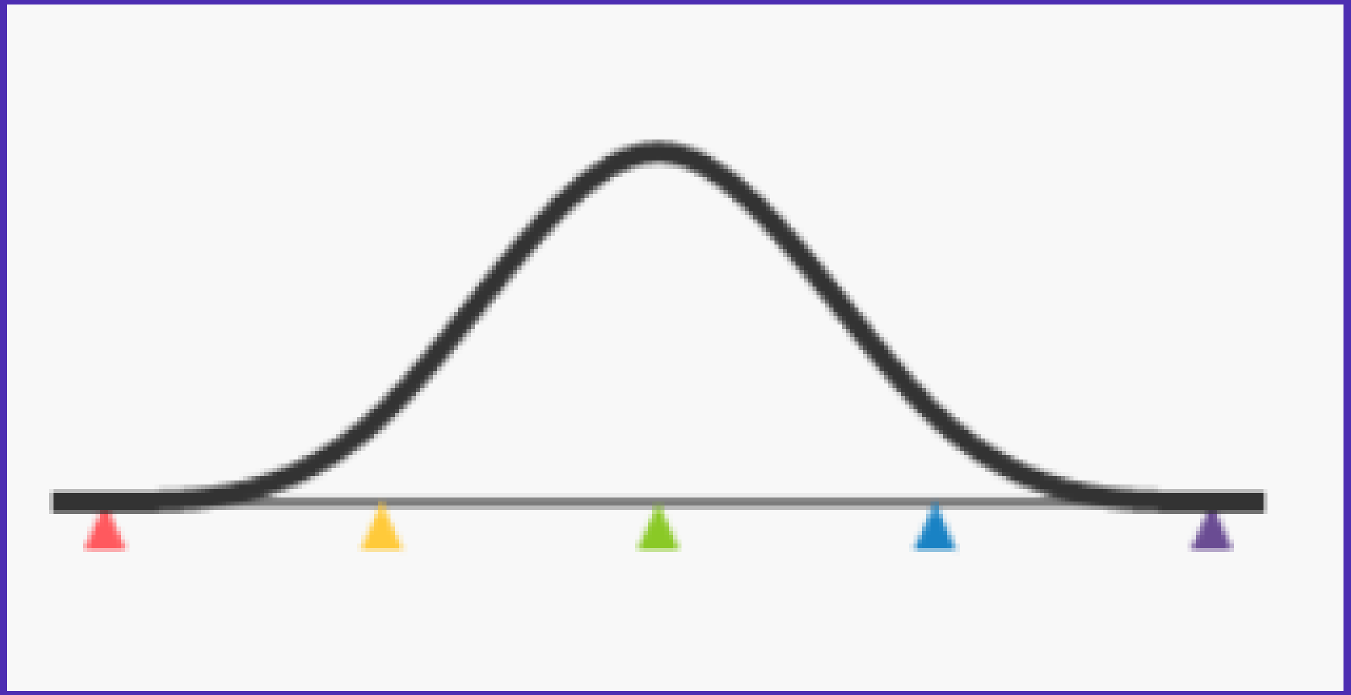
- In the last section of this series of graphs we've been looking at how different curves behave when all the control points are moved around as a group.
- The most primitive B-spline function of order 1 has a value of 1.0 in a small range and 0.0 everywhere else as seen below.
- It is generated from two lower order functions, with a simple procedure that you can witness by dragging the slider.

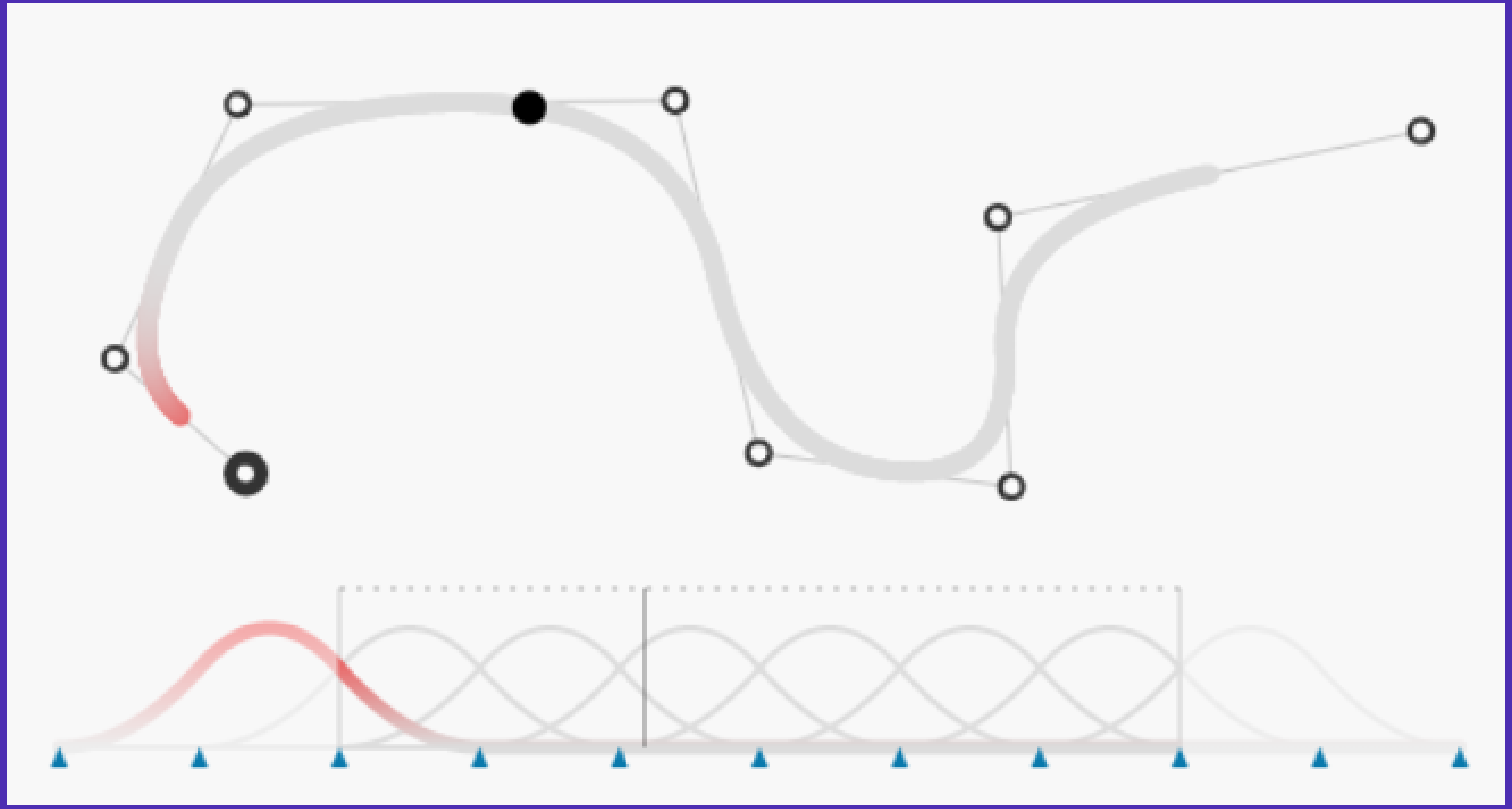




Building The Basis (Contd.)

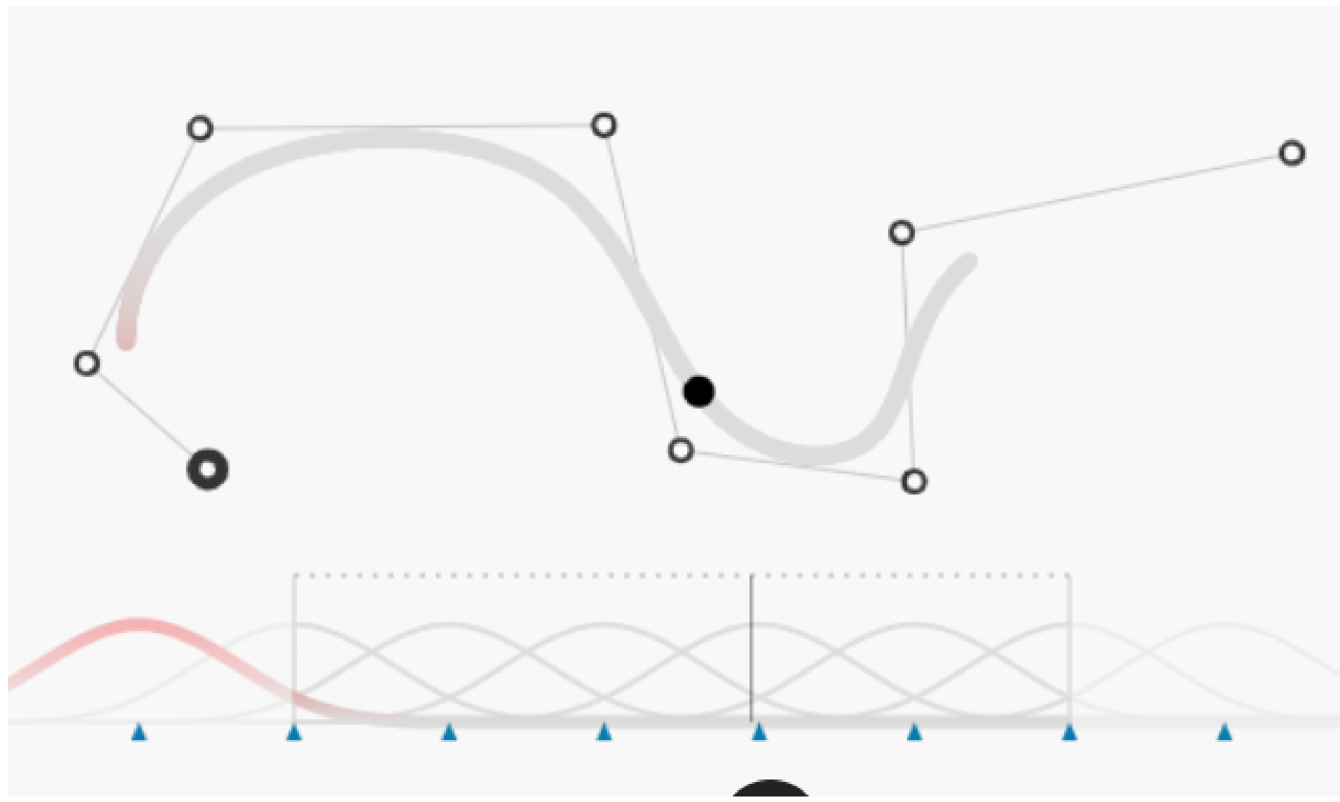
- A B-spline is a function of order 2 and its curvature is also continuous making the entire curve G2 continuous. In the diagram below you can see that the curves are designed over a repeating, uniform interval that I've been marking using small triangles at the bottom. However, nothing prevents us from adjusting those intervals as we please, making their distribution non-uniform.

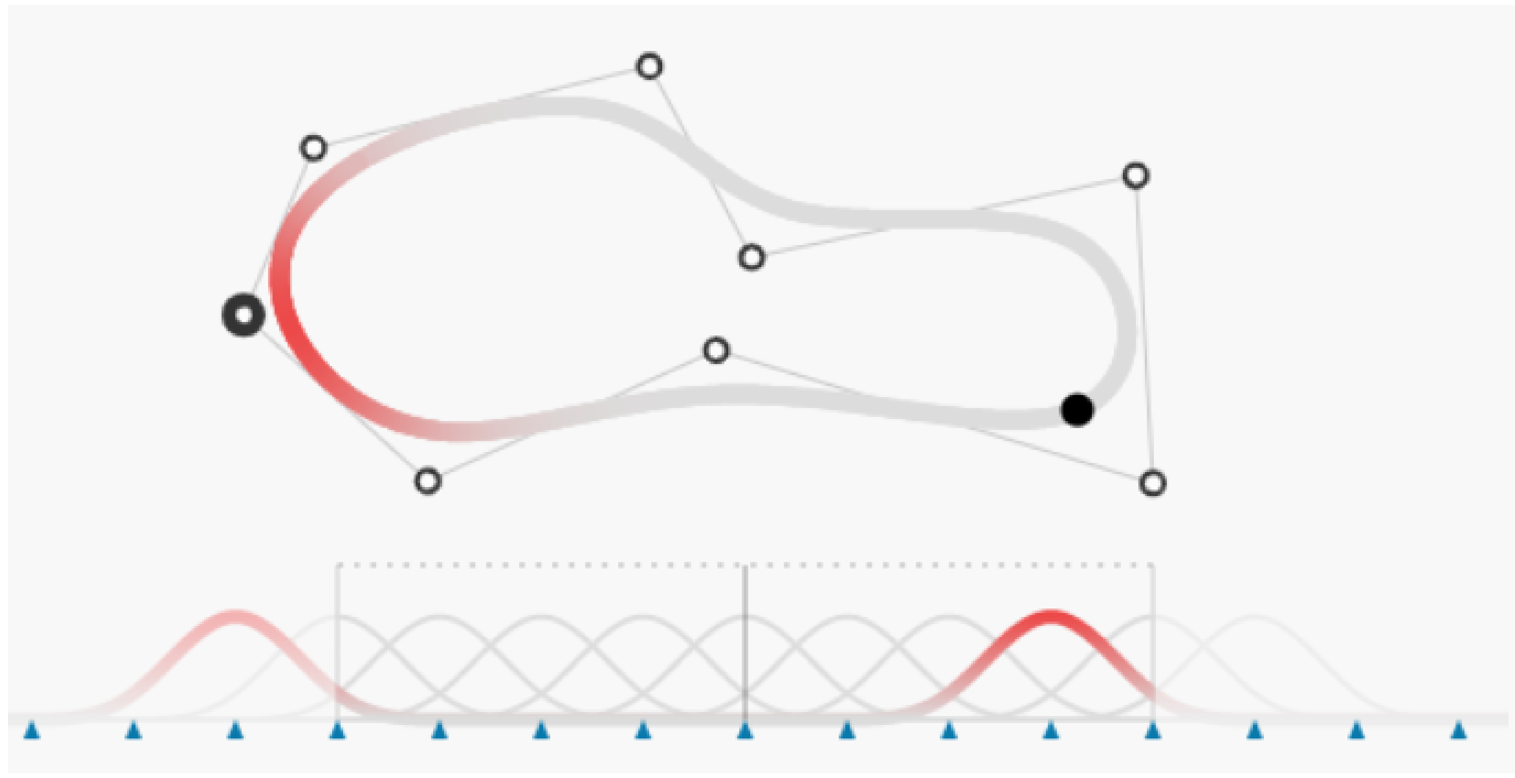




Building The Basis (Contd.)

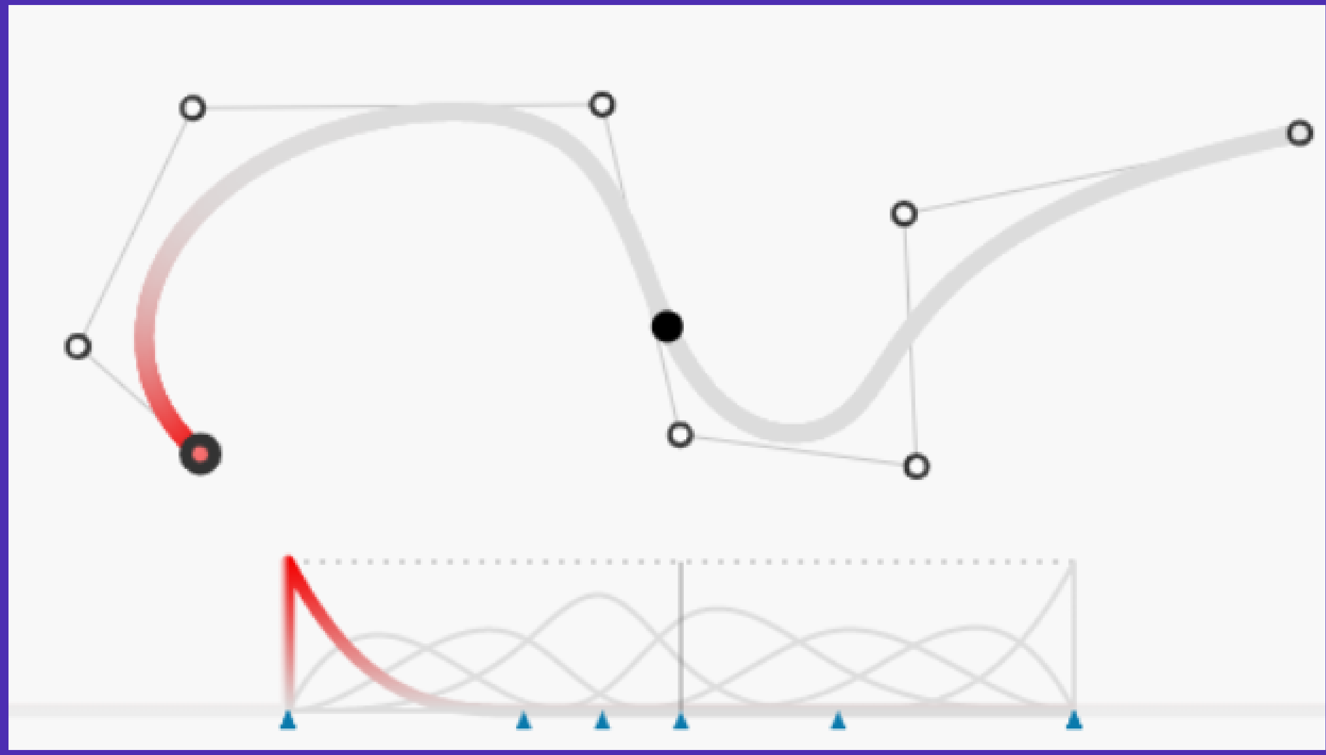
- When the first four and last four knots of a cubic B-spline curve overlap, the curve extends all the way to the boundary control points as desired. In the demonstration below you can see what happens to the curve as we perturb the position of the interior knots as well. With uniformly distributed knots we'd need to overlap three consecutive control points to get a sharp corner. By moving three knots into a single location, we can achieve a sharp corners reigned in by just a single control point.

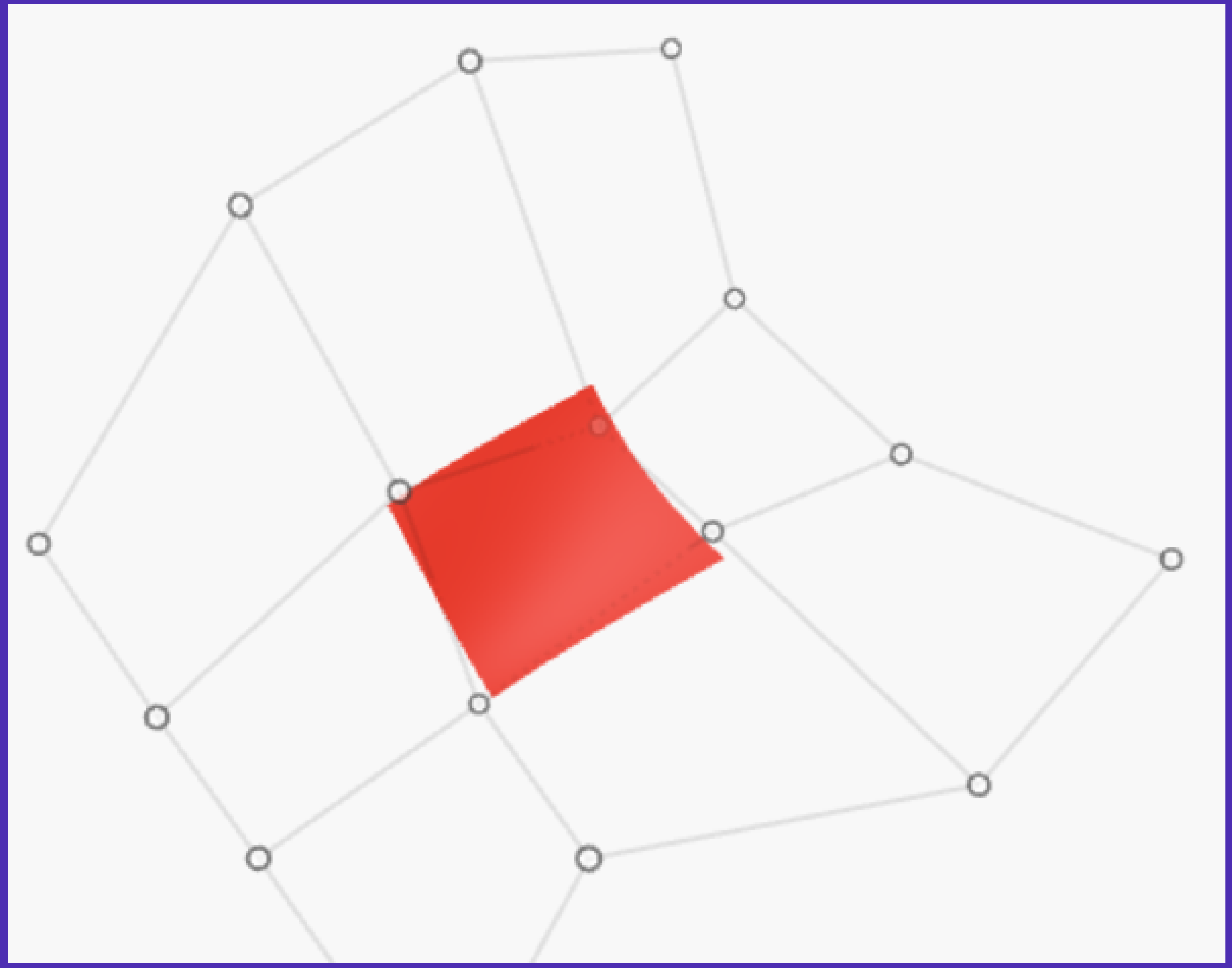




B-spline Extravaganza

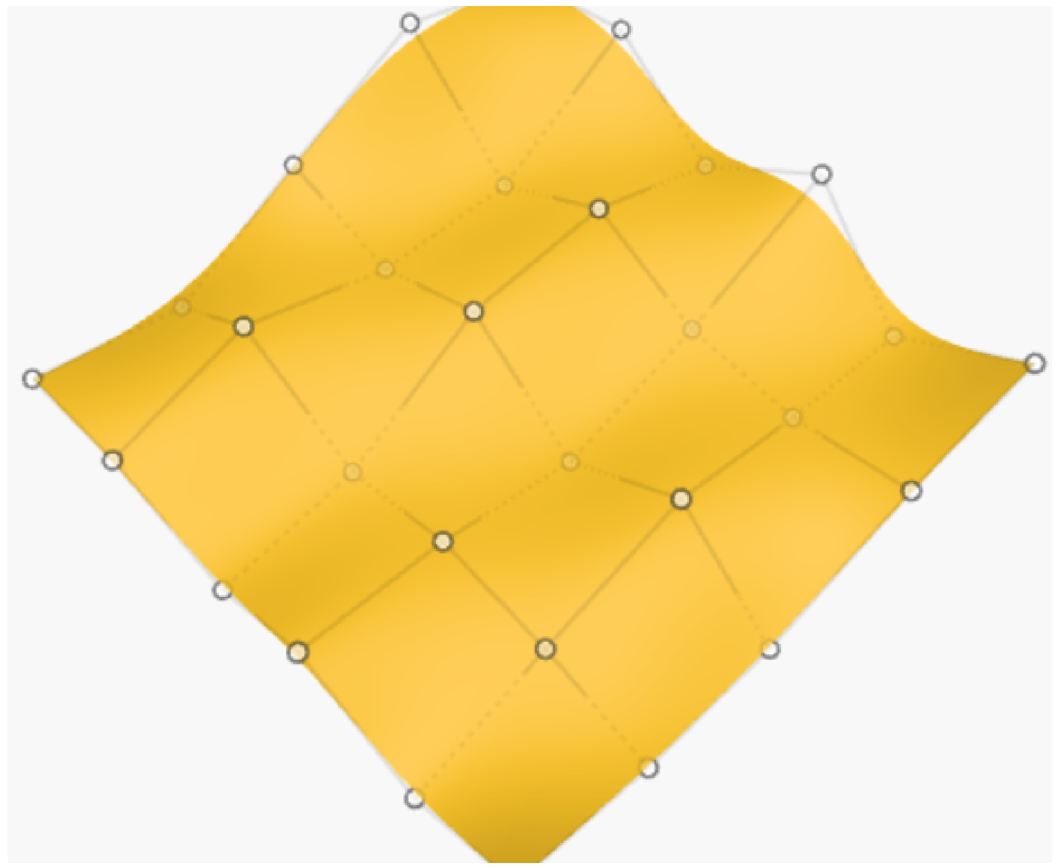
- Each of the weighting functions has been equally significant, but we can simply deviate from this assumption by shifting the relative weights of some functions to make them more essential. Consider what happens to the curve when we change the weight of the fourth control point, which I've labelled with a letter D.
- The B-spline is a non-uniform knot distribution with arbitrary weights. As we change the weight of one function the shapes of all the others are adjusted as well. This ensures that all the weighting functions add up to 1 over the entire valid range which maintains the partition of unity. By combining this with the Bézier curve we can create non-Uniform rational B-Splines, also known as NURBS.

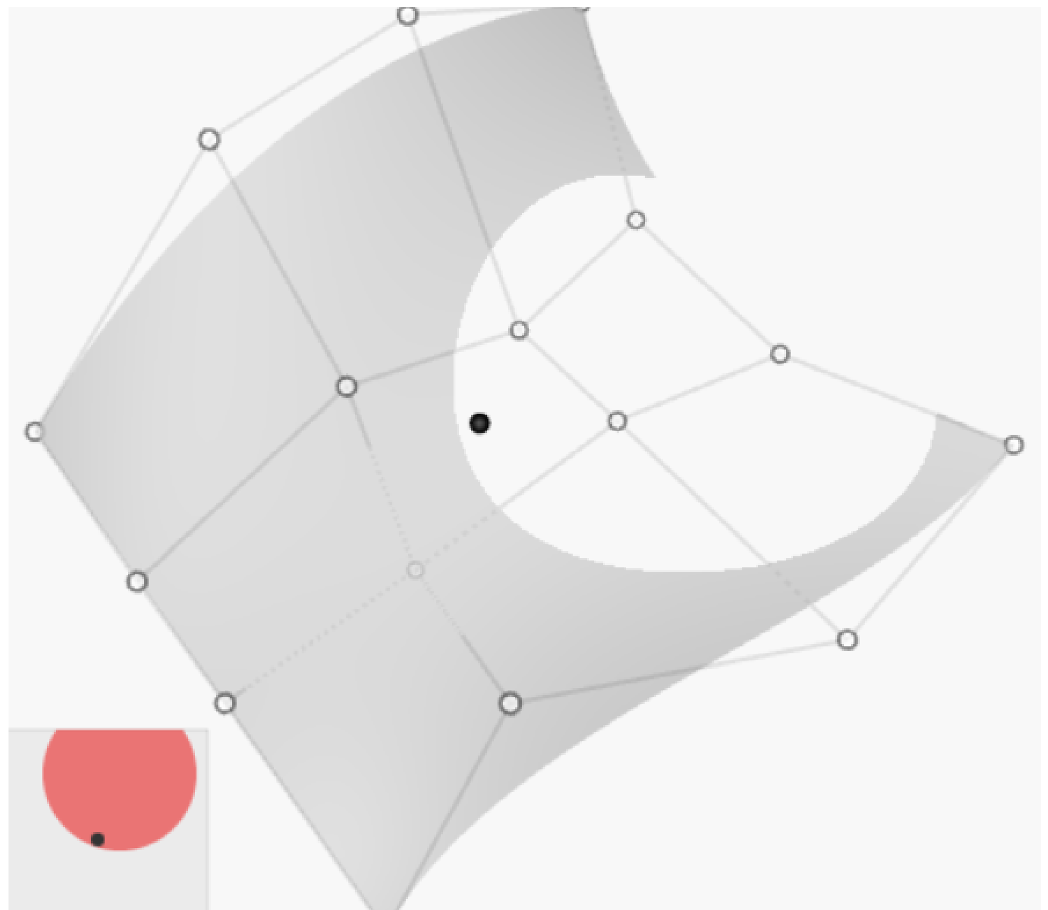




B-spline Extravaganza(Contd.)

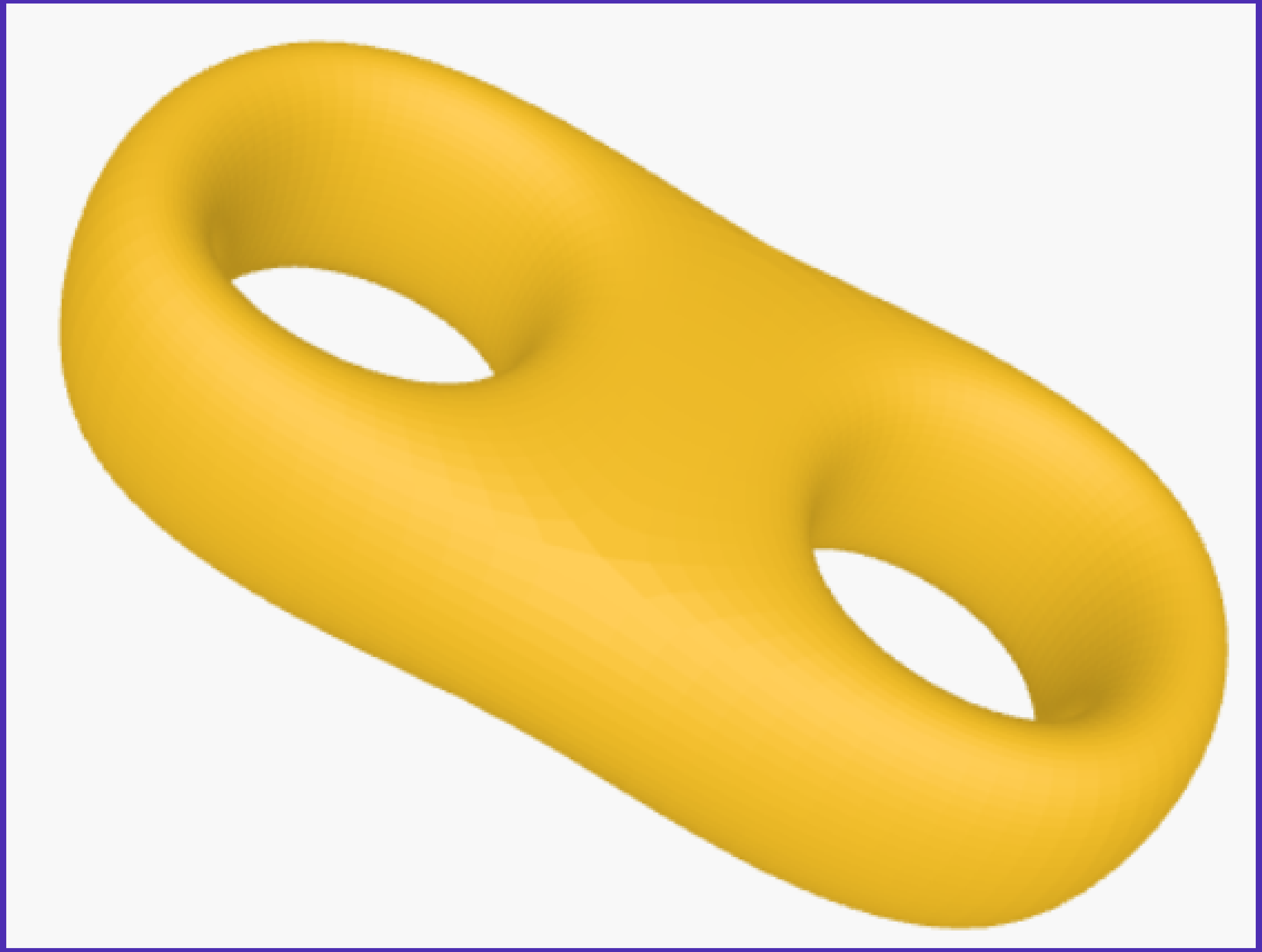
- We may also create more complex forms by using larger rectangular control point grids. We can make the surface go all the way to the control mesh's boundaries by adjusting the position of the knots.
- NURBS curves and surfaces are very powerful and they have a lot of flexibility, but they still have their limitations when it comes to modeling shapes. One approach is to simply ignore some parts of the input square. In the demonstration below you can control the radius of the red circle that defines the region where we're not drawing the surface.





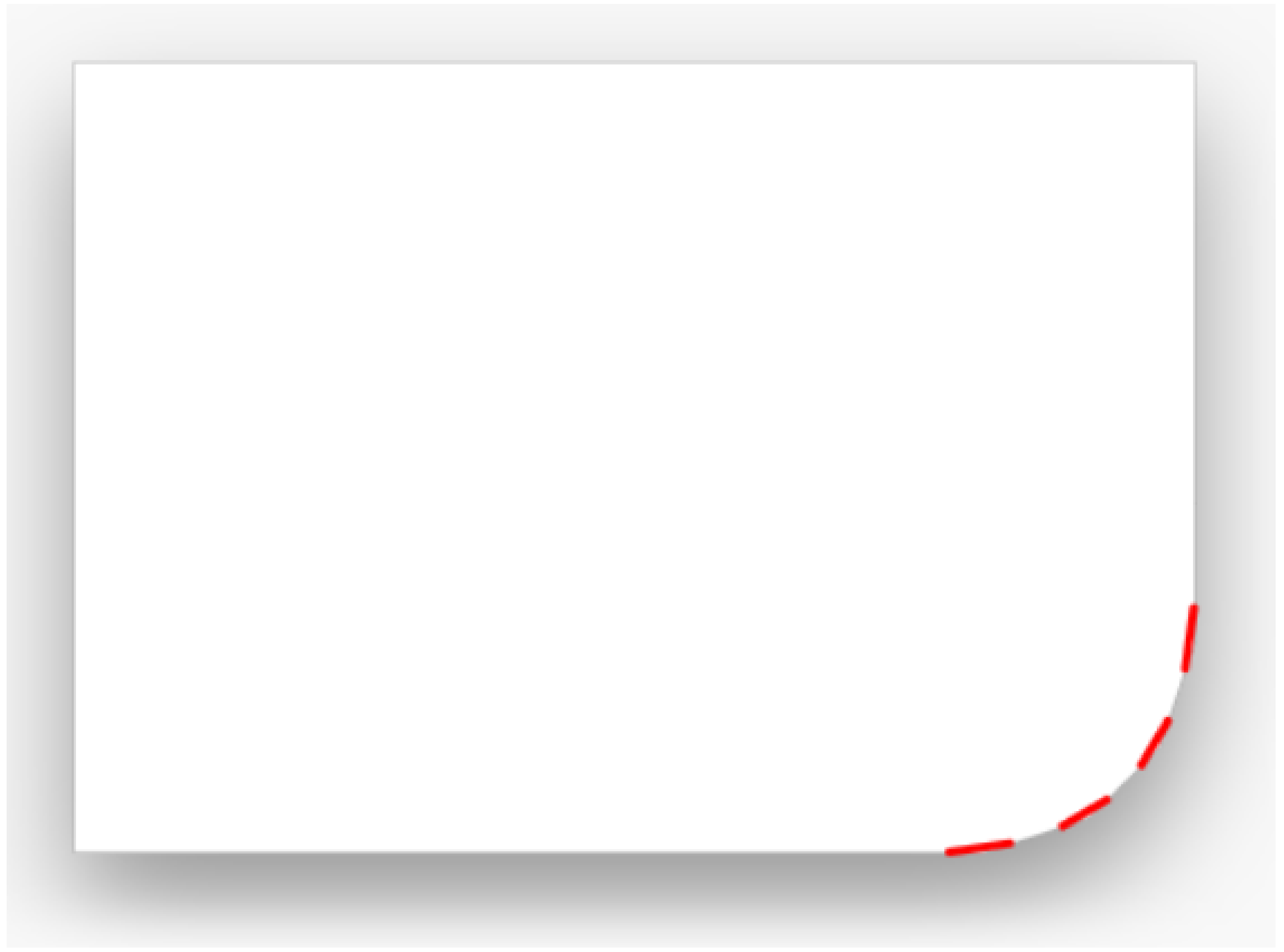
B-spline Extravaganza(Contd.)

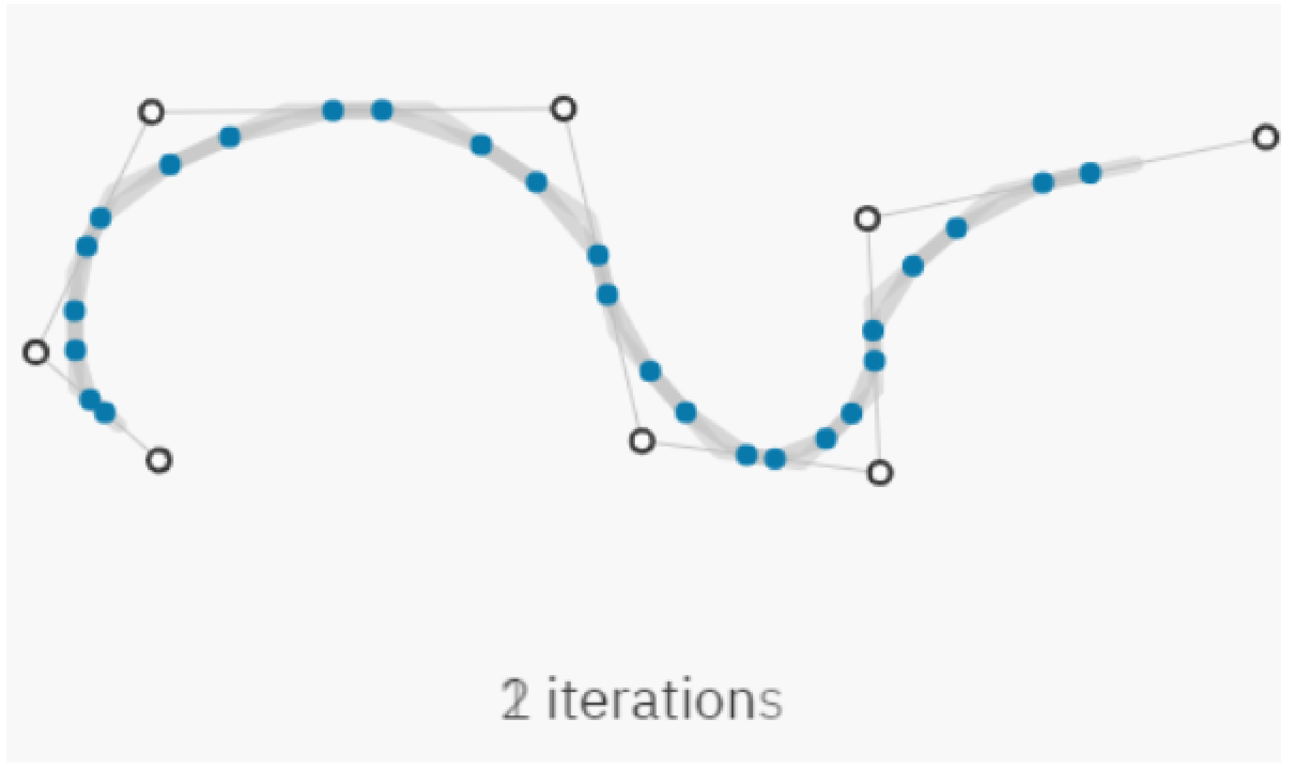
- Because getting the proper shape of the hole on the surface necessitates refining the hole's shape on the input square so that it appears right after being distorted by the surface, this step is a little time-consuming.
- You may have noticed that all the control points we've been dealing with are rectangular in nature. This heavily limits the types of surfaces we can create. For example, it's impossible to express a surface like the one below without having to somehow manually stitch it at the edges. However, the entire point of using B-spline curves and surfaces was to ensure automatic continuity and smoothness in the first place.



Cutting Corners

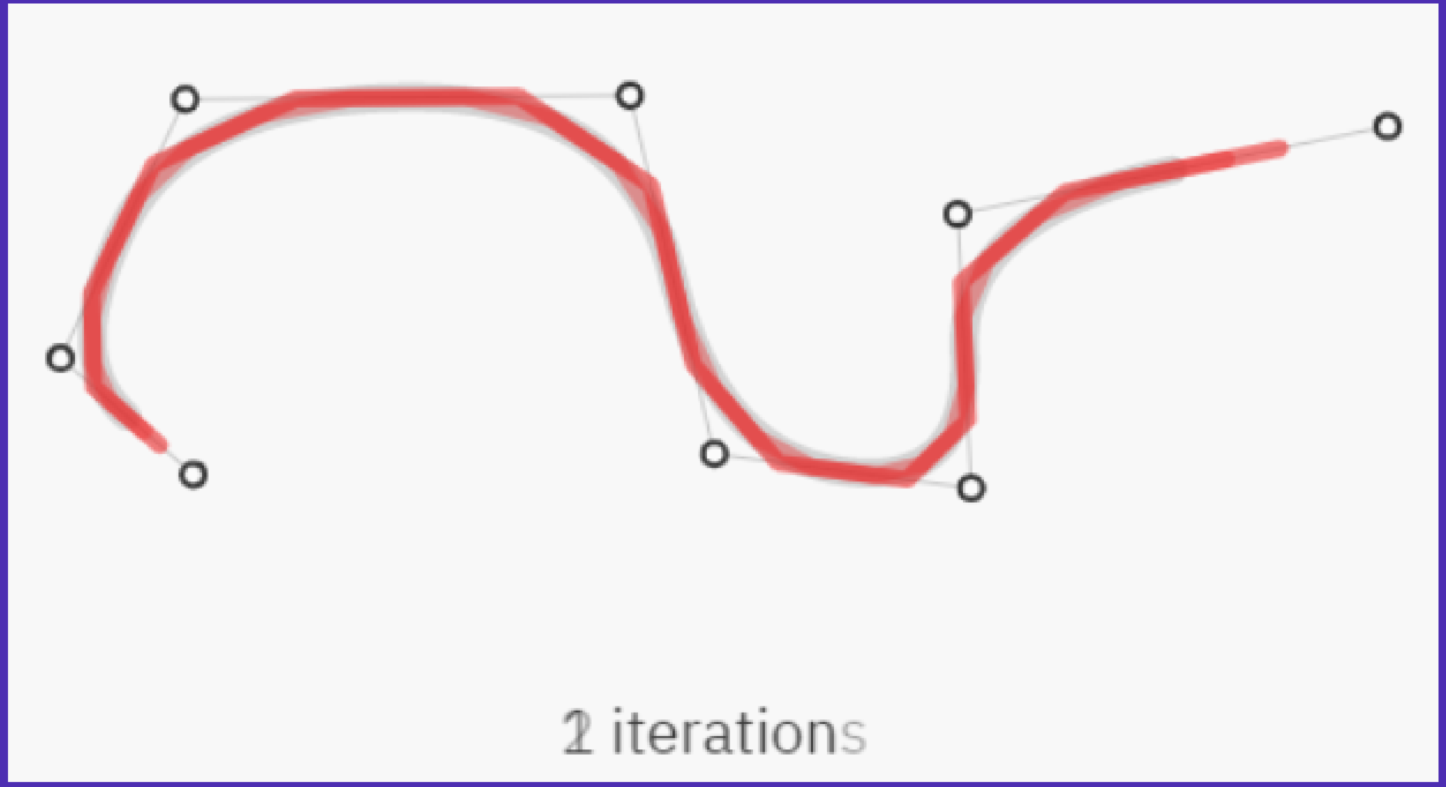
- Let's start with a blank piece of paper instead of an actual drawing board. Those are normally rectangular, but if we wanted a lovely rounded corner on a sheet of paper, we'd have to work a little more. Cutting off the sharp corners repeatedly is one of the simplest ways to achieve it.
- The corner is already very round after just three repetitions. We can also use this corner cutting technique to chip away at a curve's control polygon. You may adjust how many times we executed the corner cutting technique and the relative distance at which the cut occurs in the demonstration aside.

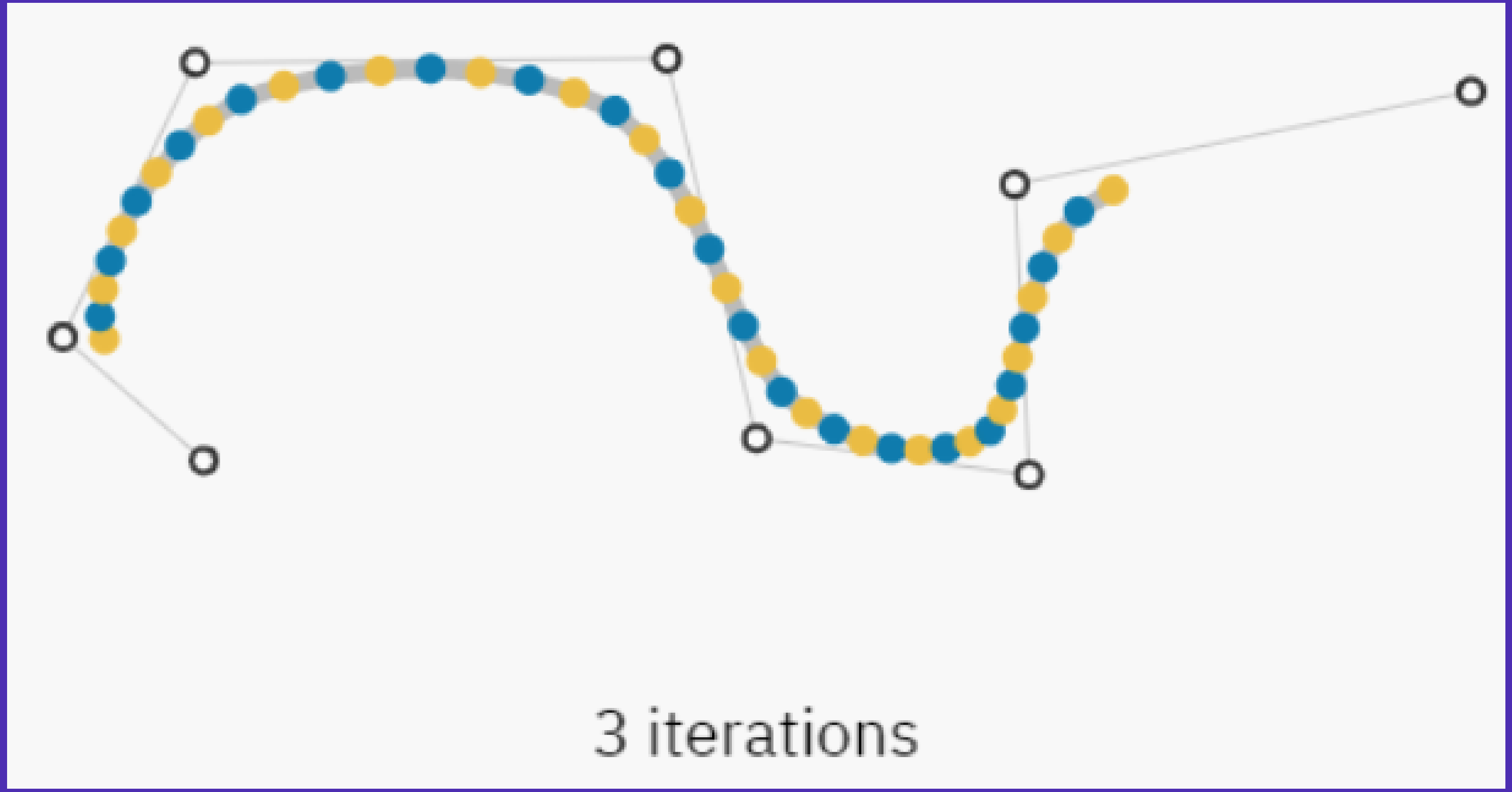




Cutting Corners(Contd.)

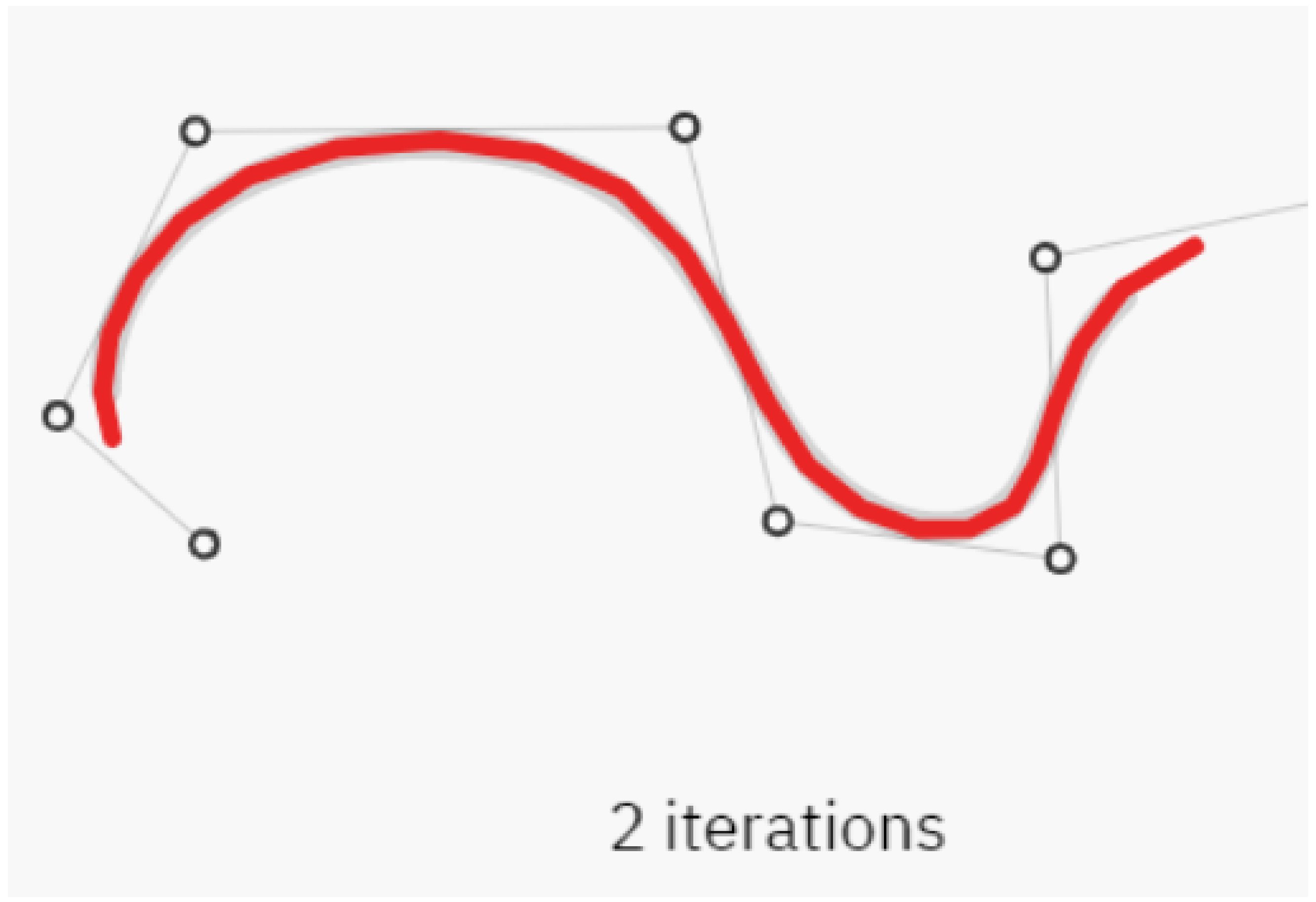
- If we cut corners too much or too little, the final polyline will look jagged and bumpy, but for intermediate values it starts to look very smooth. A particularly interesting curve is generated when the cut happens 1/4 of the way into the straight segments.
- This is because they're obtained by repeatedly cutting off corners of their control polygon. The cutting scheme we've used so far placed two new points on an existing edge, but that's not the only choice we have.

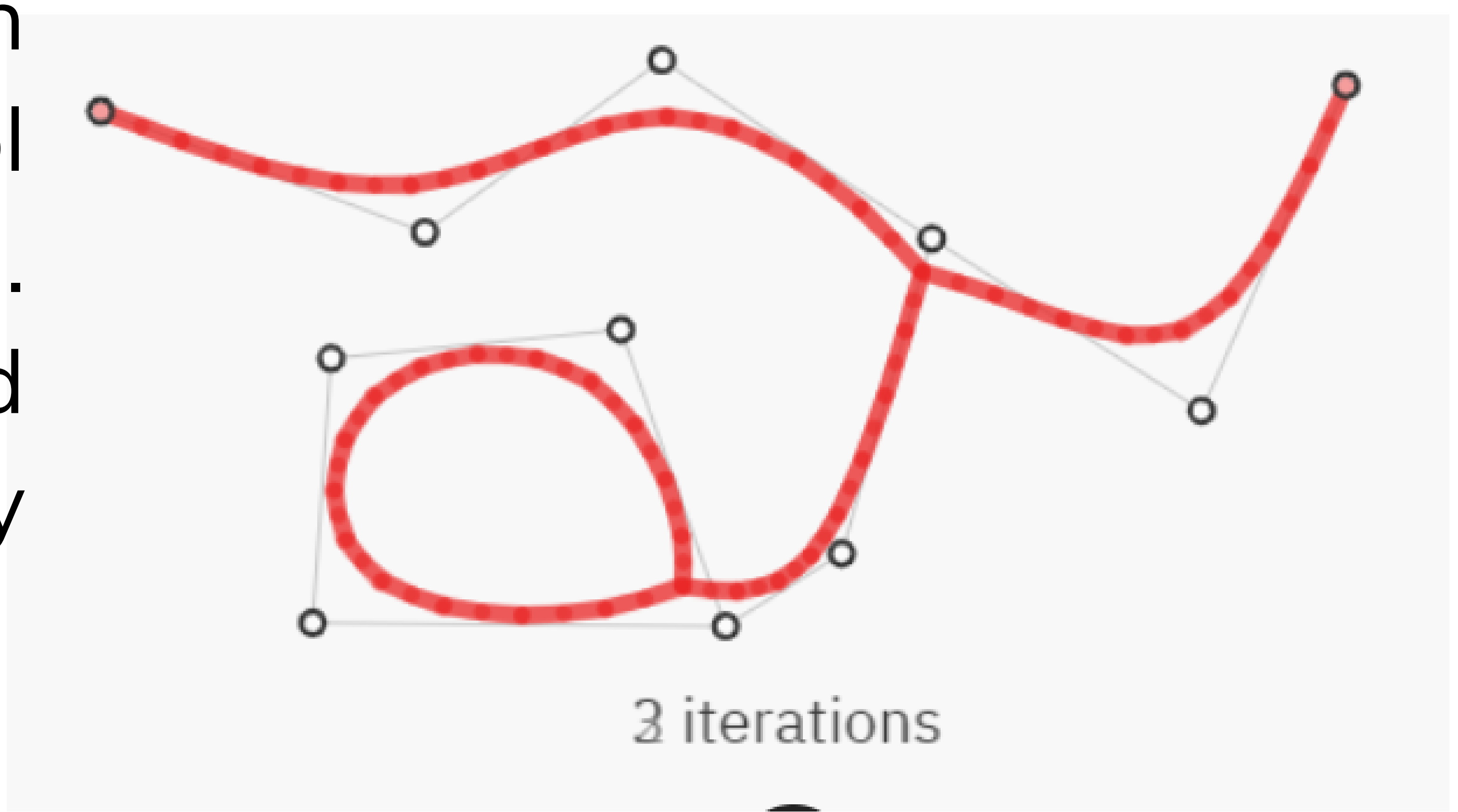




Cutting Corners(Contd.)

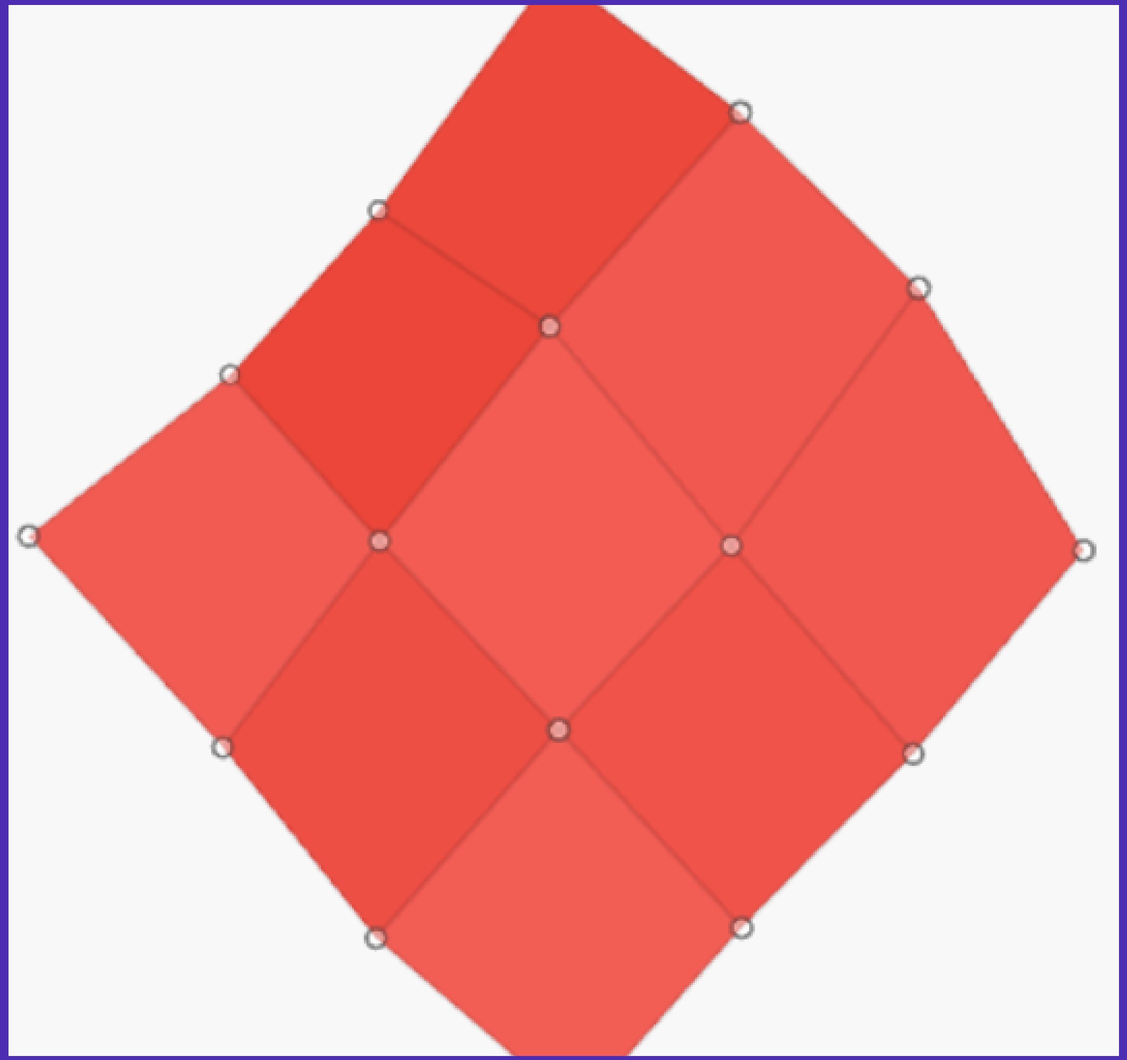
- In this scheme the new edge point. Enew lands right in the center of the edge, and the new. new vertex point Vnew is a weighted average of the original. and its previous and next neighbors Vprev and Vnext. The curve it generates in limit is a cubic B-spline curve.
- Subdivision curves extend the power of B-splines curves into more arbitrary connectivity. We can modify the subdivision rules for the "tail" control points so that their new vertices don't move at all. This makes the curve stick to its end points, and allows us to handle branches and loops very easily.

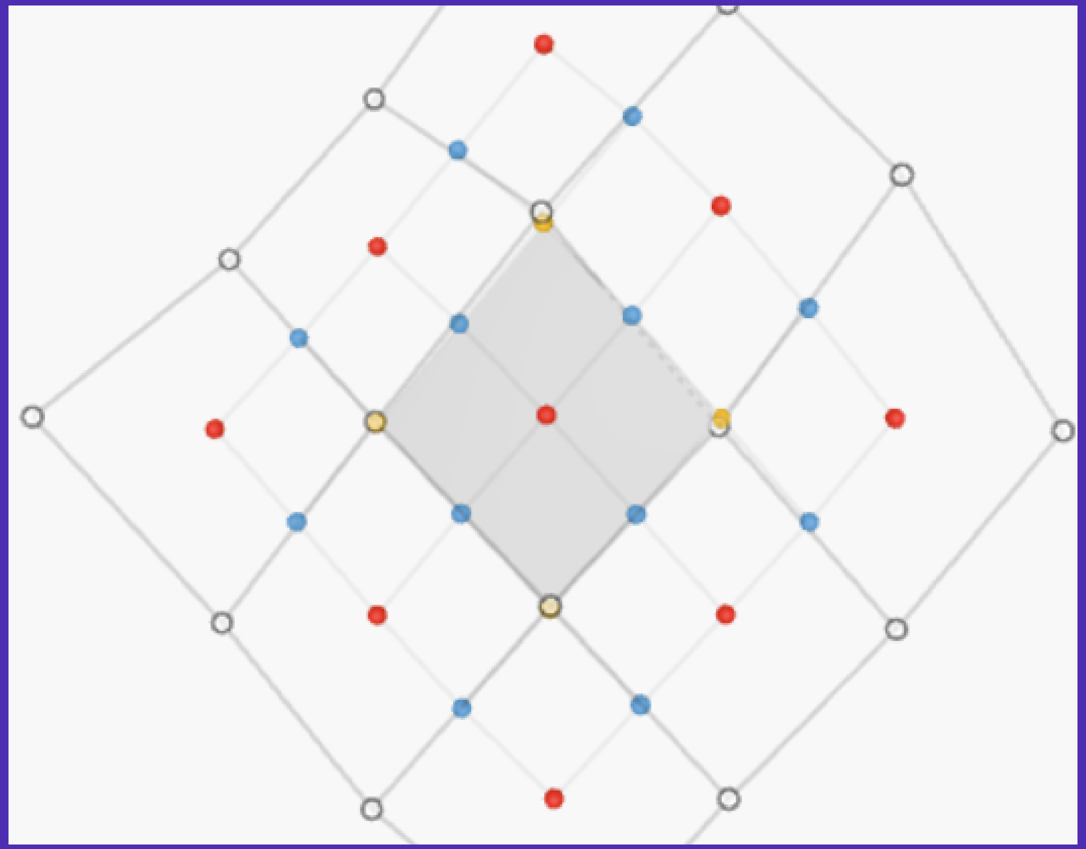




Subdivision Surfaces

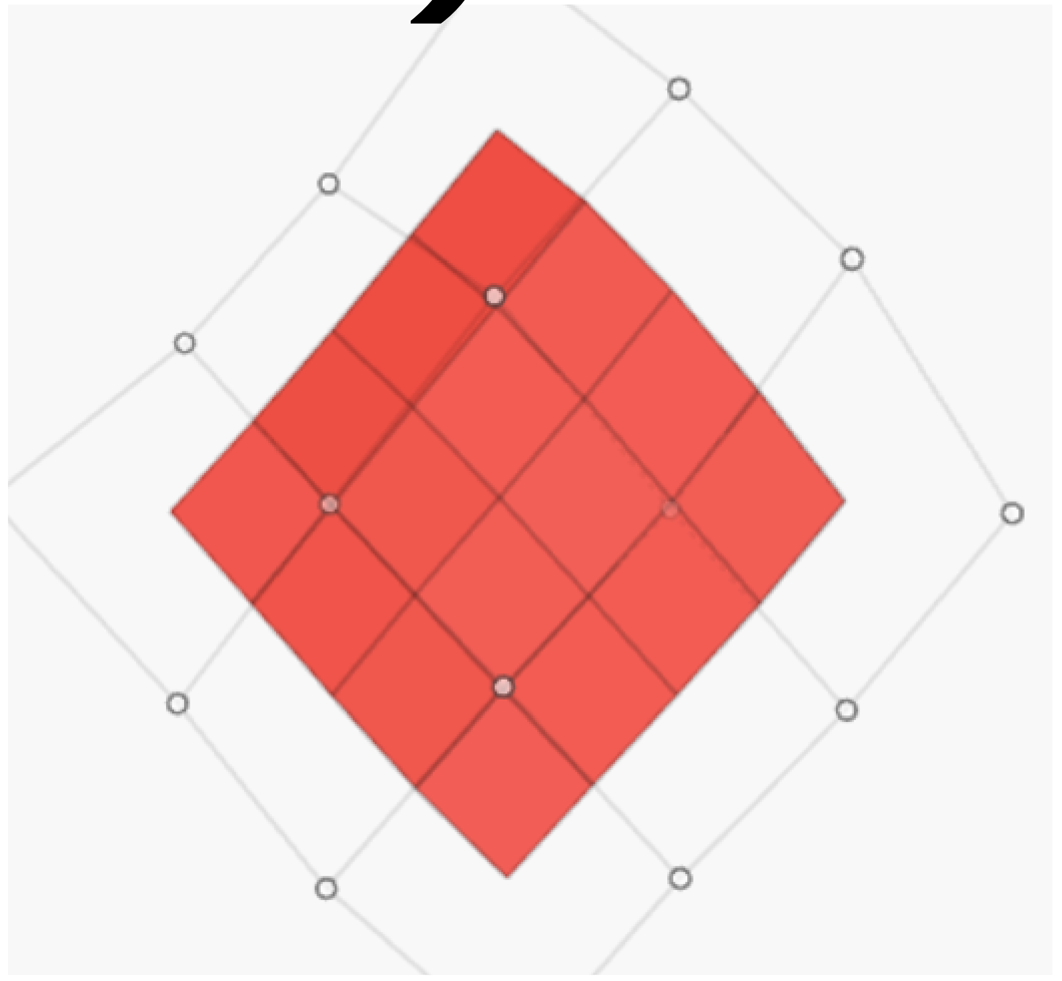
- Subdivision surfaces have been developed by Ed Catmull and James Clark. They were inspired by work by Vladimir Chaikin on corner cutting in a quadratic B-spline. After infinitely many subdivisions that surface does turn into a cubic B-spline surface.
- They started with a single cubic B-spline patch defined by a 4x4 control grid, and then searched for 25 new control points that would define a surface with the same shape, The arithmetical procedures needed to create these new control points can be classified into three distinct groups that end up corresponding to original faces, edges, and vertices.

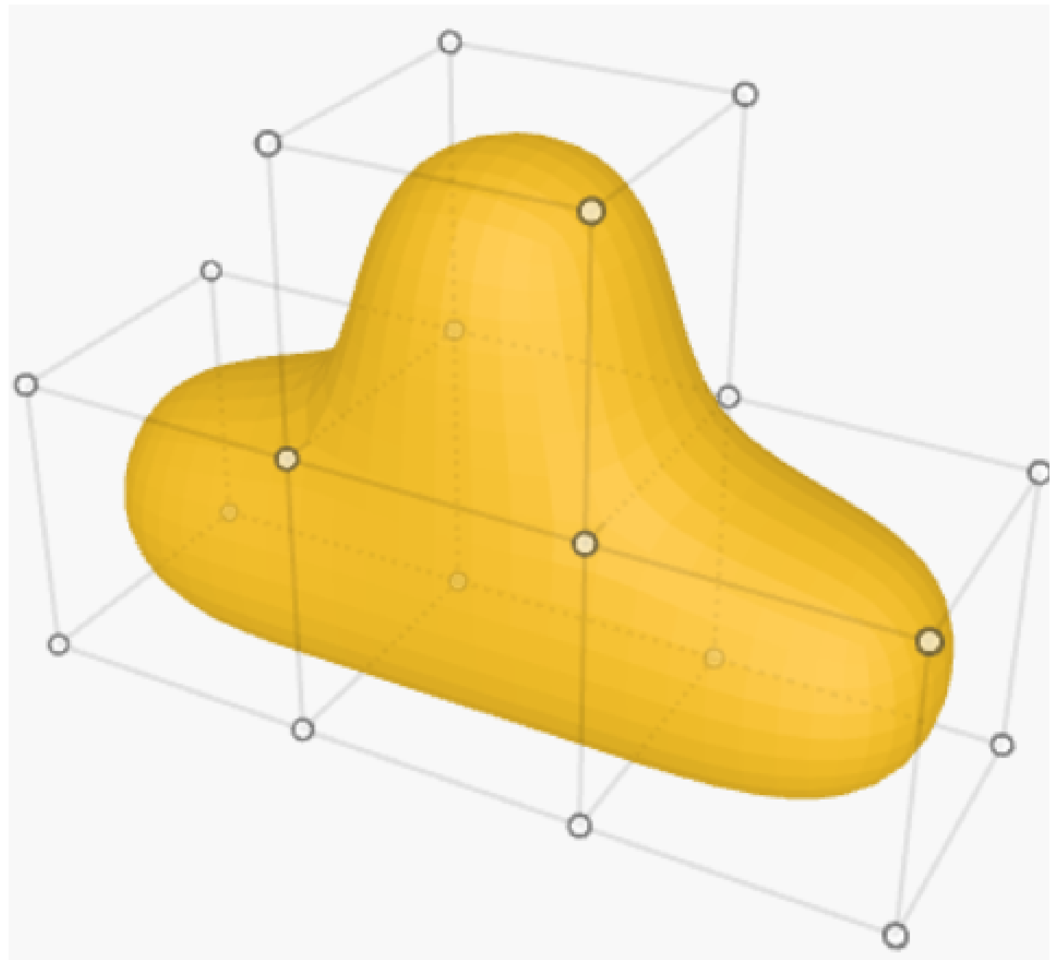




Subdivision Surfaces(Contd.)

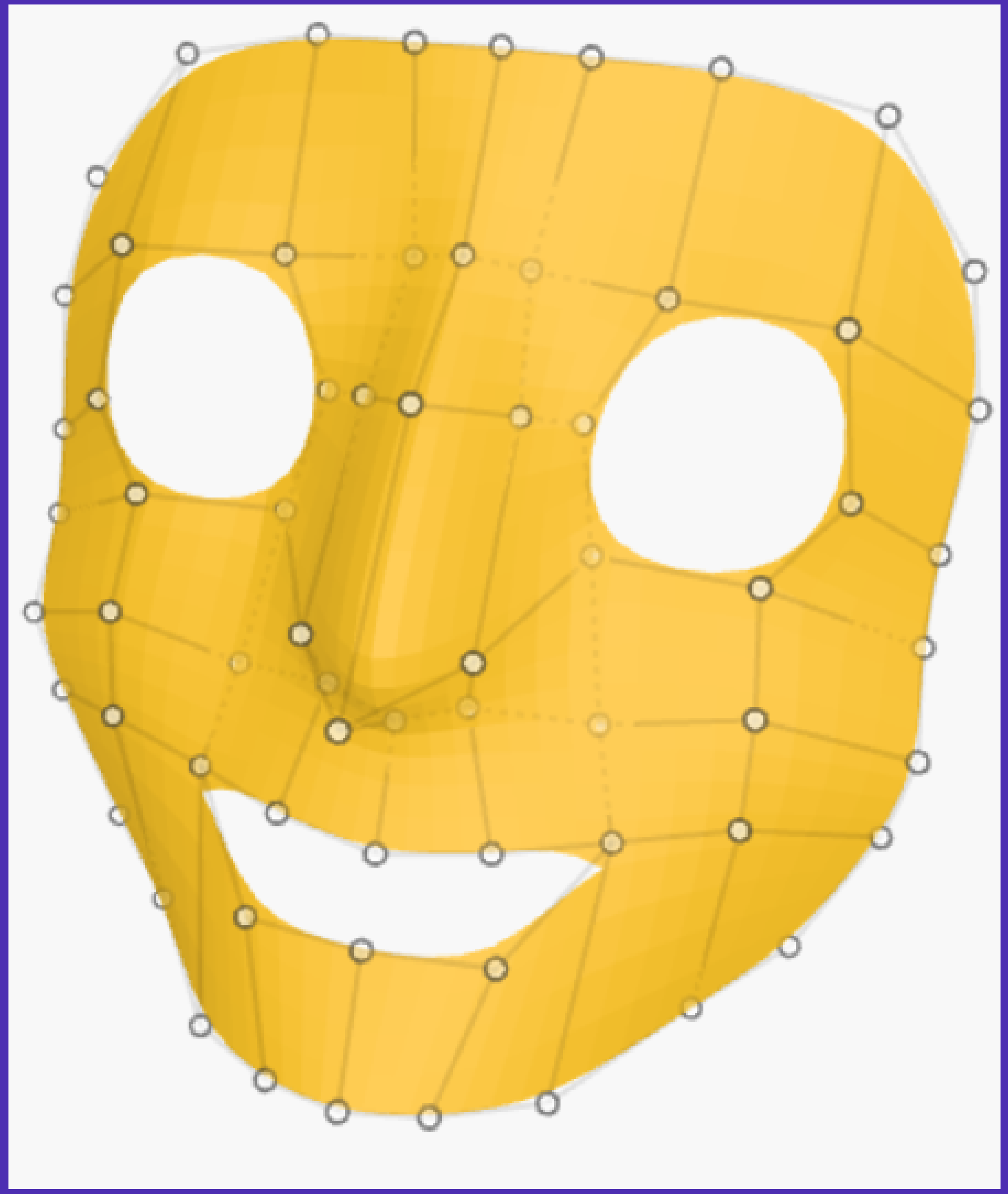
- The Catmull-Clark subdivision scheme also works when the vertices of the initial mesh have different number of neighbors – in the mesh below each vertice has 3, 4, or 5 neighbors. If we repeat that scheme over and over again the created faces of the control mesh will actually create a cubic B-spline surface. The crucial invention here is that nothing about the rules for creating the new subdivided control mesh relied on how the mesh itself is connected.

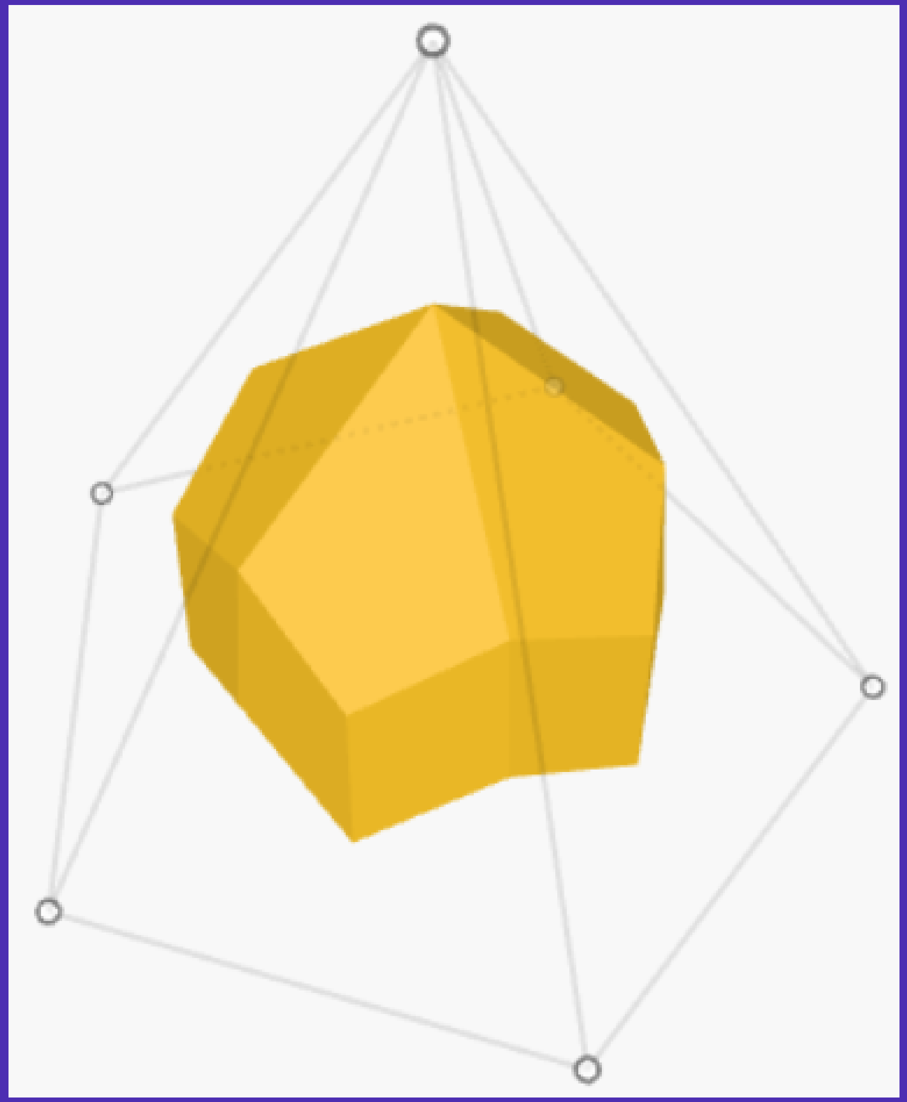




Subdivision Surfaces(Contd.)

- Subdivision surfaces make it very convenient to describe perfectly smooth surfaces using relatively simple meshes. To express holes in a subdivision surface all we need to do is to not create faces in some regions of its defining mesh. The initial faces also don't necessarily have to be four-sided, the subdivision rules also work for meshes whose faces are triangles or more complicated polygons..





THANK YOU

Contact Me

+91-9872913054

f20180047@goa.bits-pilani.ac.in