Curve and Surface Constructions

Milind Sohoni
http://www.cse.iitb.ac.in/~sohoni

Introduction

This talk is about the creation of curves and surfaces. These are created by the kernel, say as:

- Projected Curves
- Surface-Surface intersections.
- Extrudes.
- Blends.
- Offsets.

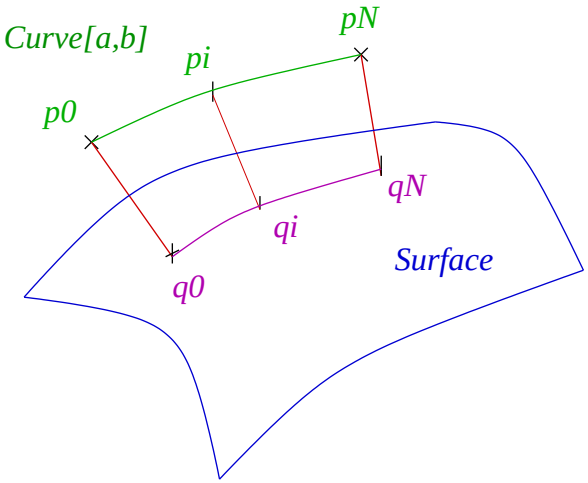
As opposed to the earlier, which was about the creation of points with special properties, this is about the construction of higher dimensional entities such as curves and surfaces. The basic paradigm is:

1. Create Points in a systematic fashion.
2. String them up as control points of a curve/surface.

Projected Curves

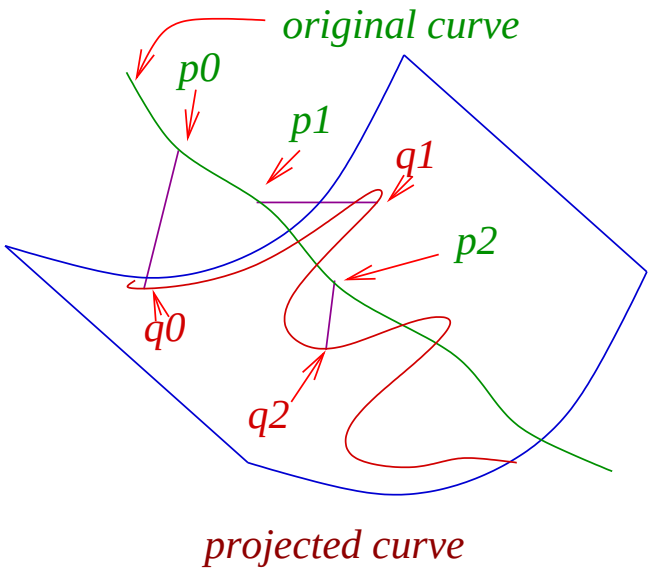
Let C be a curve and S be a surface. The projected curve $proj(C, S)$ consists of all points $q \in S$ obtained by projecting points $p \in C(t)$ onto S. The simplest constructor for this is seemingly straight-forward:

1. Depending on tolerance requirements, choose a Δ.
2. For $a + n\Delta \in [a, b]$, project $C(a + n\Delta) = p_n$ to get q_n.
3. Define C' with control points $q_0, \ldots, q_N$.



Not So Simple....

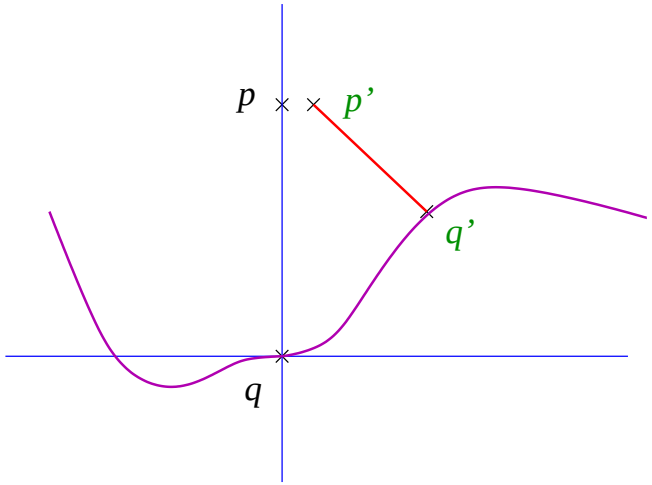
Here is a bad case: The curve C is close to the axis of the almost cylindrical surface. As a result the projected curve is highly unstable.



This situation has to be detected and the user alerted.

Detection: Rudiments of Curvature

Lets tackle the point-projection on a curve, which too has the same instability.



Let C be a curve, and $p = (0, h)$ be a point so that the point $q = (0, 0)$ is the closest point to p on C. Furthermore, let us assume that C is given by

$$y = \frac{a_2}{2!}x^2 + \frac{a_3}{3!}x^3 + \dots$$

Now we exmaine the point $p' = (\epsilon, h)$ and see how $q' = (x_0, y_0)$ moves.

Perturbation Analysis

The equation is the the line $p'q'$ is perpendicular to the tangent at q'. Given the equation defining C, we see that:

$$\frac{dy}{dx} = a_2 x + \frac{a_3}{21} x^2 + \dots$$

The perpendicularity condition translates to:

$$(a_2 x_0 + \frac{a_3}{21} x_0^2 + \dots) \cdot (\frac{y_0 - h}{x_0 - \epsilon}) = -1$$

Simplifying this, and substituting for y_0, we get:

$$(a_2 x_0 + \frac{a_3}{21} x_0^2 + \dots) \cdot (\frac{a_2}{2!} x_0^2 + \frac{a_3}{3!} x_0^3 + \dots - h) = \epsilon - x_0$$

Finally...

Re-organizing:

$$(a_2 x_0 + \frac{a_3}{21} x_0^2 + \ldots) \cdot (\frac{a_2}{2!} x_0^2 + \frac{a_3}{3!} x_0^3 + \ldots - h) = \epsilon - x_0$$

We see that:

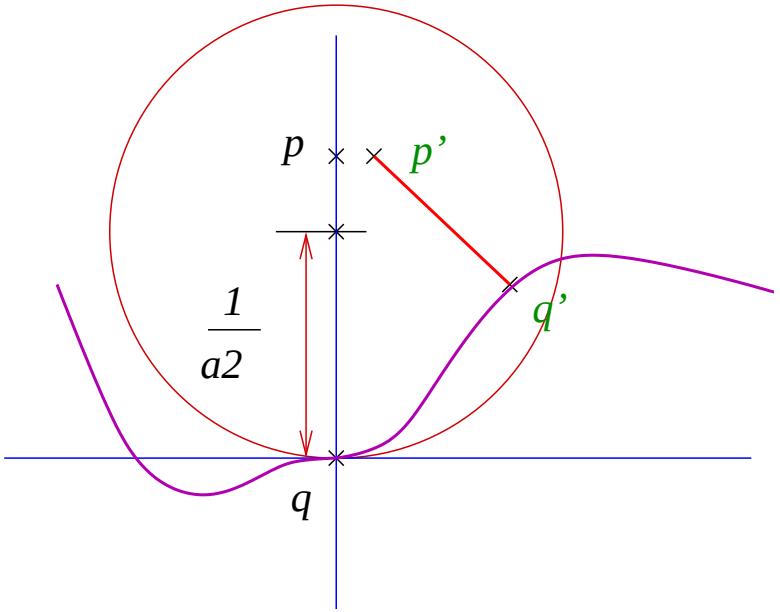
$$(1 - a_2 h) x_0 + \frac{h a_3}{2!} x_0^2 + \ldots = \epsilon$$

By classical analysis, this can be inverted iff the linear term is non-zero i.e., $(a_2 h - 1) \neq 0$. The inverse is:

$$x_0 = \frac{1}{1 - a_2 h} \epsilon + \ldots \text{ higher terms}$$

The constant a_2 is called the curvature of the curve at the point q and $\frac{1}{a_2}$ as the radius of curvature.
We see that if p is normally away from q *exactly* distance $\frac{1}{a_2}$ away, then the projection is unstable.

And Pictorially

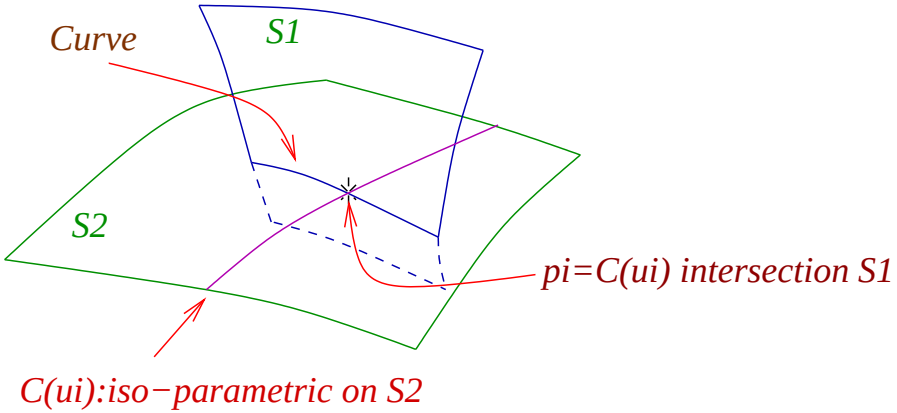


The detection is done by computing the constant a_2 and comparing $\frac{1}{a_2}$ with h. Note that $a_2 = f''(0)$ is the double derivative.

Surface-Surface Intersection

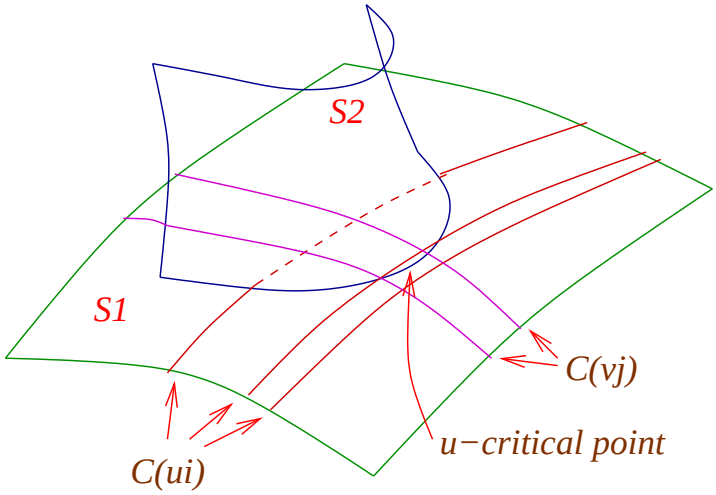
Let S_1 and S_2 be two surfaces. we are required to find the curve of intersection. The strategy:

1. Select a sequence of parameter values u_i for S_2.
2. Construct iso-parametric curves $C(u_i)$ and intersect with S_1 to construct the point p_i.
3. String up p_i into a control polygon.



Issues

- Start of parameter value, its range, and the increment Δ.
- Constructions of iso-para curves.
- Delicate issue: Critical Points

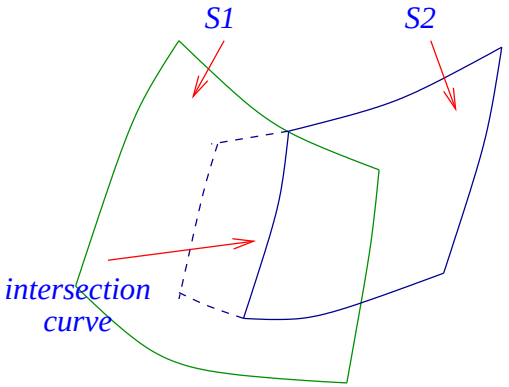


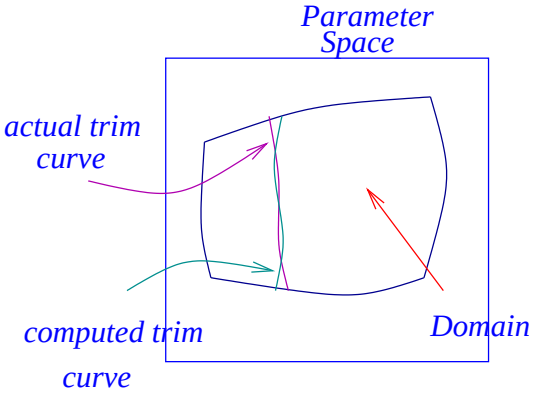
Thus such critical points have to be discovered[a] and the parameter must be moved from u to v.

[a] How is this discovered?

Trim Curves

What happens to the surfaces after having undergone intersections?

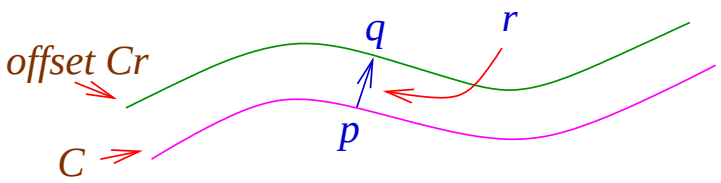




Note that the intersector throws out parameter values of the points of intersections. These are then strung up to get the trim curves.
Usually, the pre-image of the actual intersection curve, and the computed trim curve are different.

Offset Curves

Let C be a curve in a plane, and $r > 0$ be a real number. The offset C_r is the curve obtained by points which are exactly a normal distance r away from points on C.

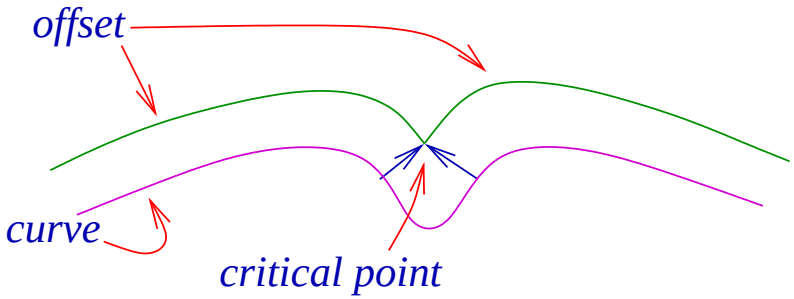


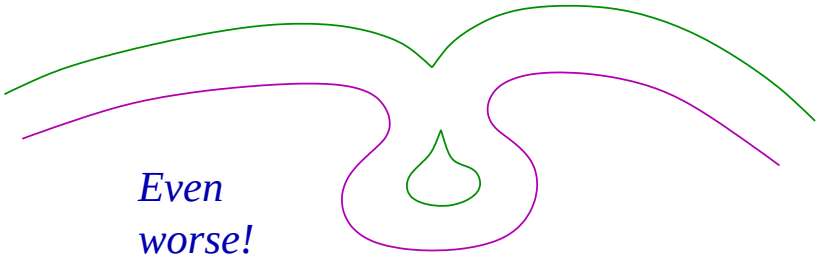
The algorithm is simple:

1. Select points p_i on the curve C.
2. Move them normally by a distance r to get q_i.
3. Construct C_r from this collection $\{q_i\}$.

However...

There are bad cases.

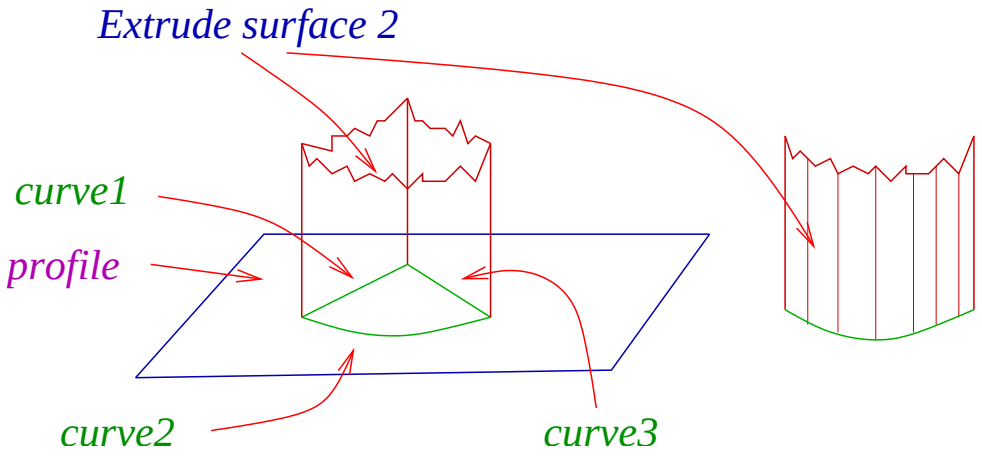




These have to be detected and the user alerted. Most kernels must support the first case above. Thus the offset of 1 smooth curve may result in 2 curves which do not meet smoothly.

Now to Surfaces: Extrude

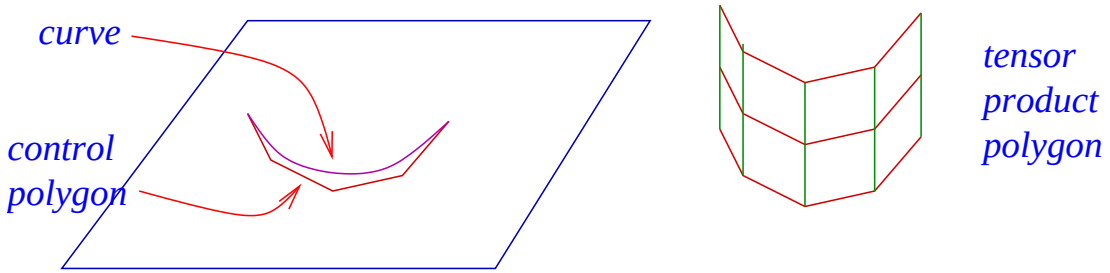
Recall that the extrude is defined by a profile P on a plane. The surface is obtained by *sweeping* this profile in a direction normal to the plane.



The Construction

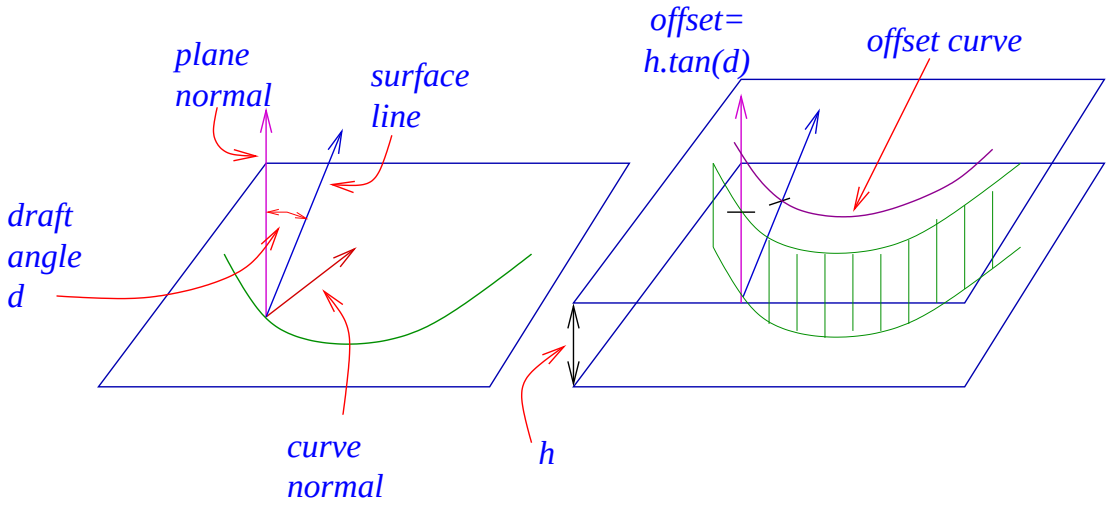
The basic steps are:

1. Unhook the profile into constituent curves and take them individually for surface geometry creation.
2. Create surface geometry as a tensor-product surface.
3. Combine these to form the *faces* of the extrude.



Drafted Extrudes

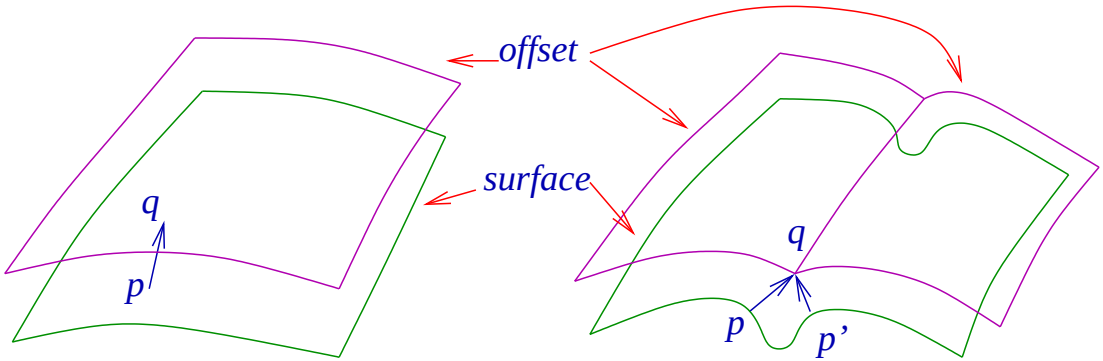
Frequently, the extrude has to be drafted by an angle d for manufacturing ease.



This causes the surface to lose the nice tensor-product construction. Now, for every point on the profile, we construct the surface line as shown. Points are taken on this line in a systematic manner.
Also note that these are also obtained by a sequence of offsets.

Offset Surfaces

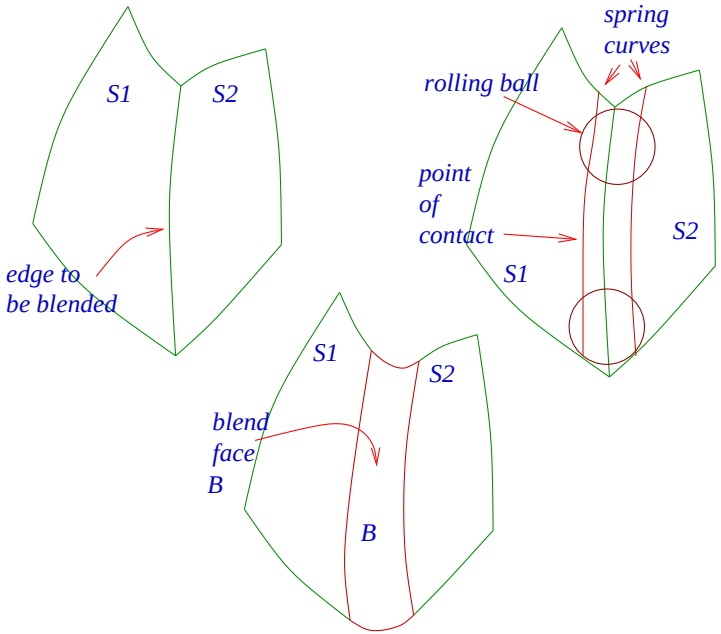
These are analogous to offset curves. If S is a surface and $r > 0$, then S_r, the offset, is the locus of the center of a ball of radius r rolling on S.
These are constructed easily by picking points on $p \in S$ and moving then normally by a distance r. There are the usual critical points which need to be handled.



The Surface Offset *and the usual problem*

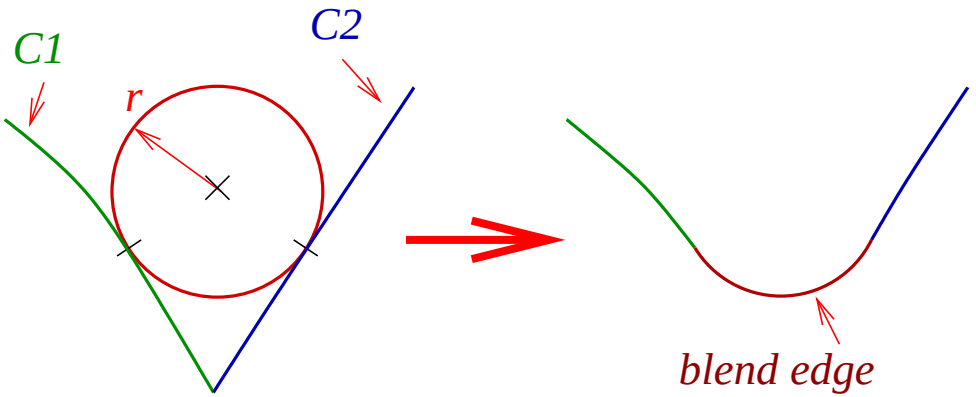
Blends

This is one of the most sophisticated of surface constructions.
The blend on the edge between surfaces S_1 and S_2 is defined as the envelope of a rolling ball which is in contact with both S_1 and S_2.



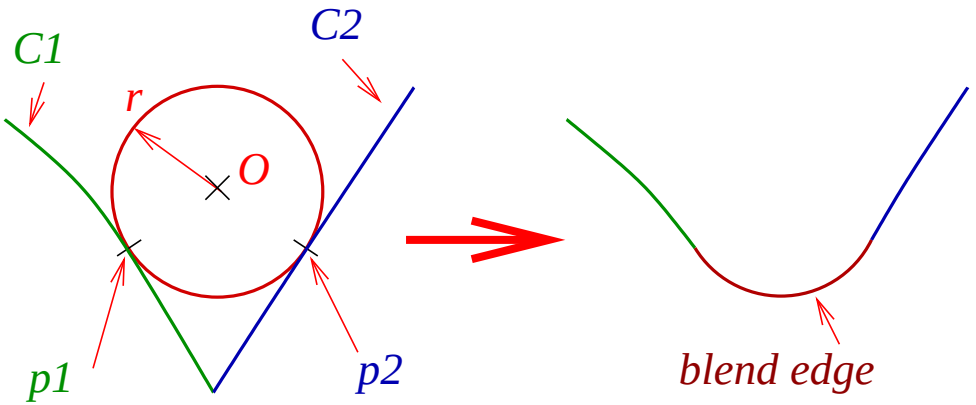
The 2D case First

So let C_1, C_2 meet at a sharp vertex, which needs to be blended with radius r.



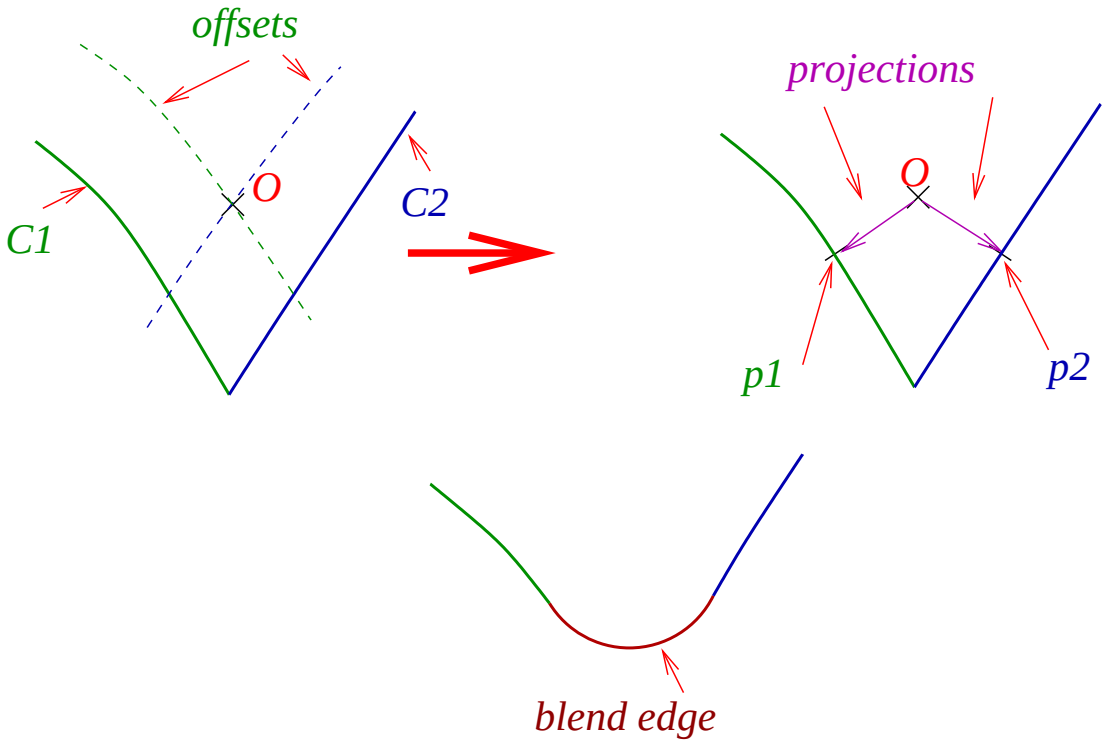
The 2D case First

So let C_1, C_2 meet at a sharp vertex, which needs to be blended with radius r.



The first task is to determine the points of contact p_1 and p_2 and the centre O of the circle.
Point O is first computed by intersecting the offsets of C_1 and C_2.

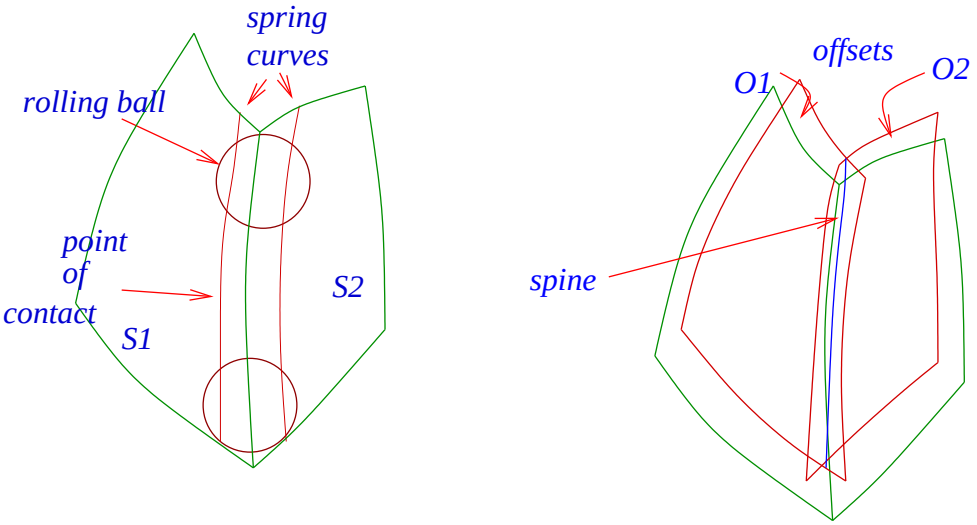
Finally



Having determined O, getting p_1 and p_2 are easy:
They are just the projections of O on C_1 and C_2. Now,

- C_1 and C_2 are truncated at p_1 and p_2.
- A circular segment with centre O, radius r is fitted.

The 3d Case

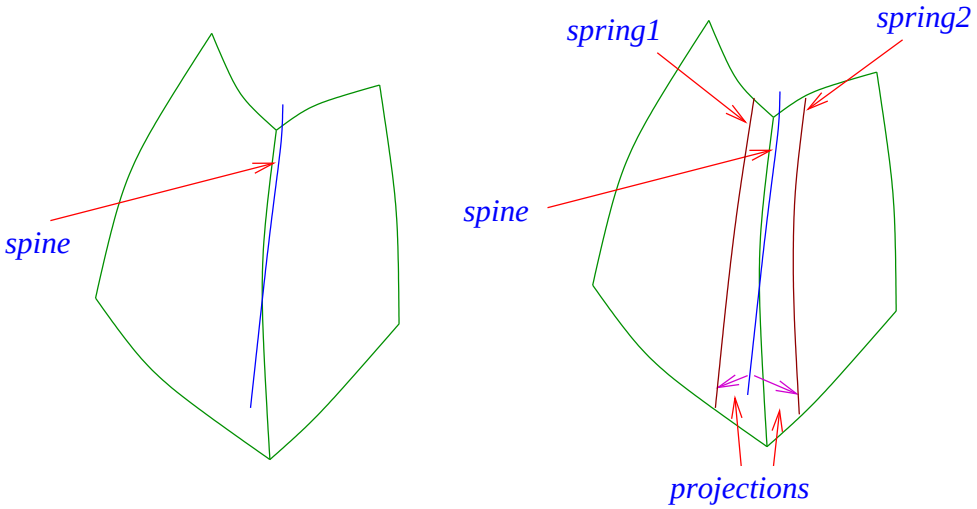


The locus of the point of contacts are called the spring curves. The blend face B is inserted between the spring curves.

The process is:

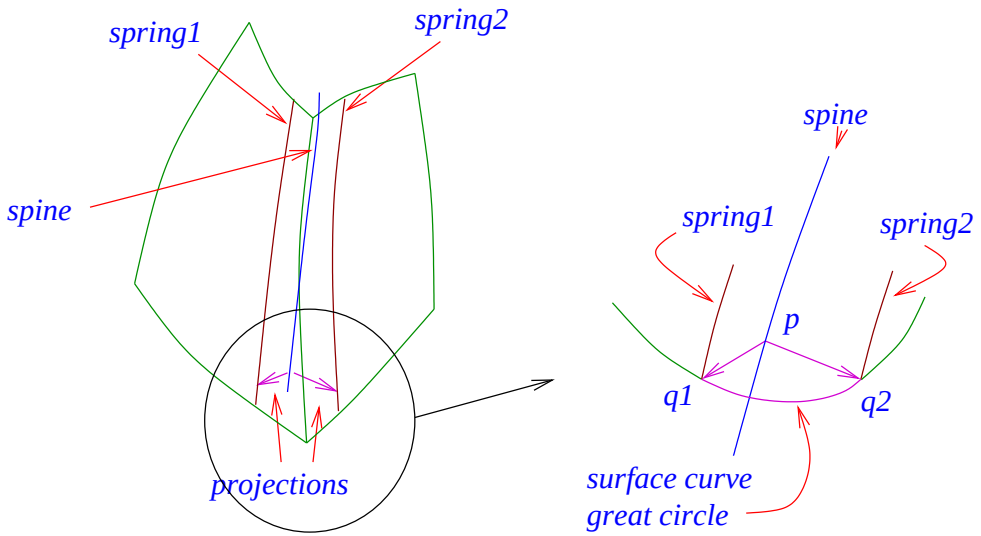
1. Define the spine to be the locus of the center of the rolling ball. This spine is computed as the intersection of the offsets O_1 and O_2 of the surfaces.

The Spring Curves



2. Compute the Spring Curves as the projections of the spine on S_1 and S_2. Prune S_1 and S_2 upto the spring curves.

The Surface Curves



3. The blend surface B is created by identifying circles which lie on B. Every point p on the spine determines, by projection, two points q_1 and q_2. The great circle is the one which (i) lies in the plane defined by p, q_1 and q_2, (ii) has p as its center, and (iii) passes through points q_1 and q_2.

Surface Construction

Thus we see that surface construction usually involves first identifying curves which must lie on the surface, and then stringing them up into the complete surface.

Identification of surface curves is a tricky matter.

Surfaces created via this paradigm include:

- Sweeps
- Revolves

And those which are NOT:

VERTEX BLENDS