Geometric Modelling Summer 2018

Prof. Dr. Hans Hagen

http://hci.uni-kl.de/teaching/geometric-modelling-ss2018

Offset Curves

Offset Curves

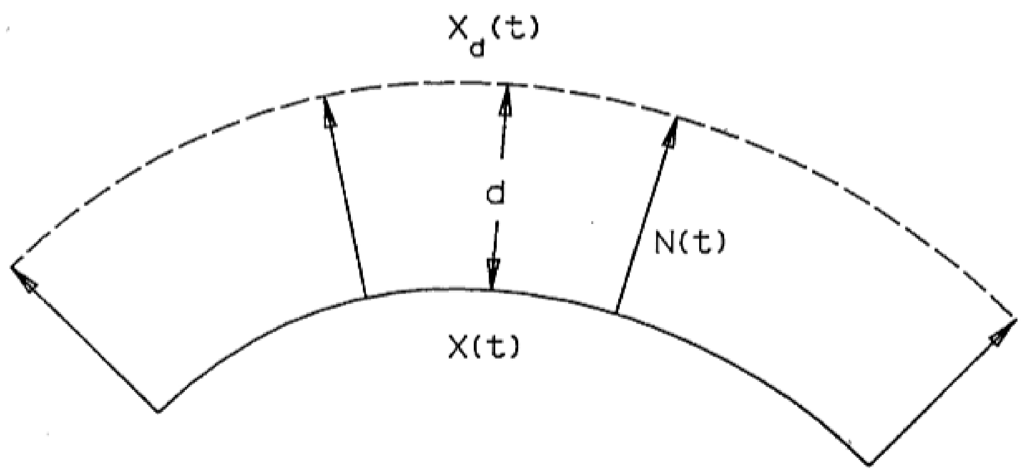
Offset curves and offset surface play an important role in numeruos applications, e.g. the steering of CNC milling machines or the modelling of surfaces with realistic material thickness.

Note: All the references in this chapter and also the techniques described here can be found in Hoschek, Lasser: *Grundlagen der geometrischen Datenverarbeitung (2. Aufl.)*; Teubner, 1992

Offset Curves

Sarting with a curve $X = X(t)$ parametrized over $[a, b]$, one can parameterize the **offset curve** X_d with distance d along the normal vector $N(t)$ as follows:

$$X_d(t) = X(t) + d \cdot N(t)$$



Offset Curves

In the case of a space curve, one can replace the "planar" normal vector $N(t)$ by the principal normal vector $e_2(t)$. Due to the normal vector, it is obvious that the offset curve of a spline curve is in general not a spline anymore.
Therefore, one is confronted with the task to find a suitable approximation.

Curves in CAD/CAM are commonly modelled as Bézier or B-spline curves, where not only polynomial but also rational representations are used. In the context of our discussion of offset curves, we first consider the Bézier representation.

Offset Curves

Original Bézier curve:

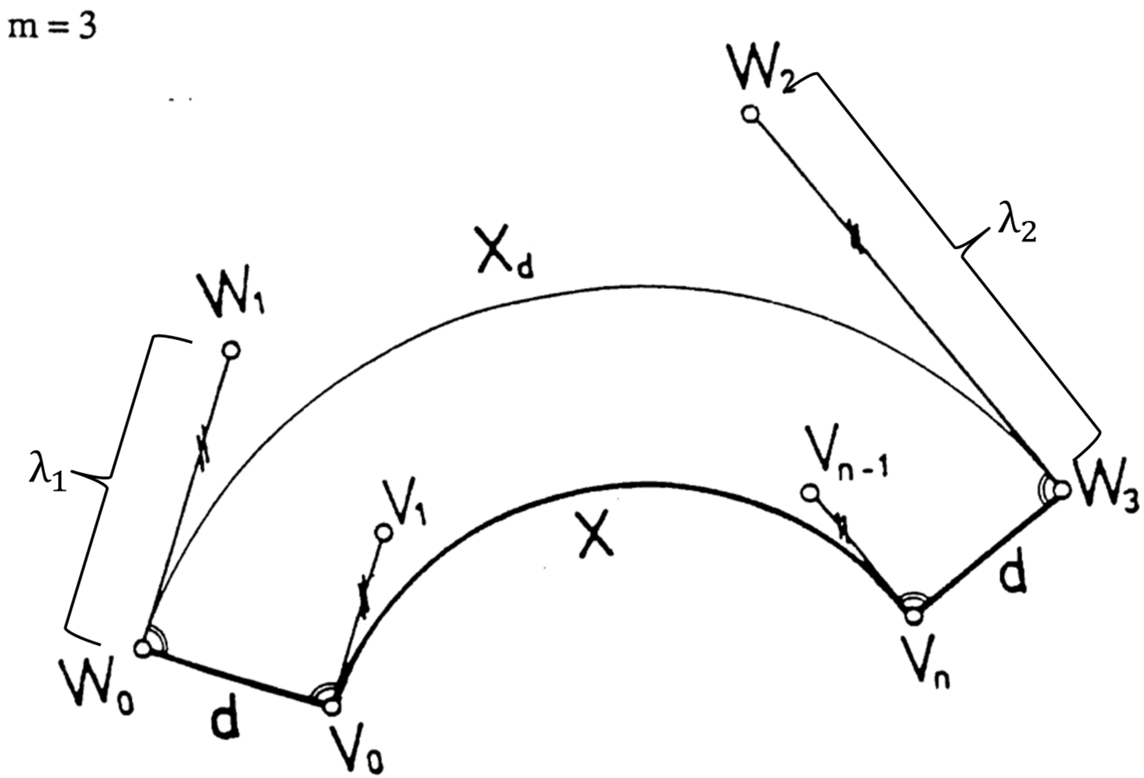
$$X(t) = \sum_{i=1}^{n} V_i B_i^n(t); \quad t \in [a, b]$$

Approximation of the offset curve X_d:

$$y(t) = \sum_{i=1}^{m} W_j B_j^m(t); \quad (m = 3 \text{ or } m = 5)$$

Offset Curves

cubic case:



Offset Curves

We have:

$$W_0 := V_0 + d \cdot N(a) \quad W_3 := V_n + d \cdot N(b)$$
$$W_1 := W_0 + \lambda_1(V_1 - V_0) \quad W_2 := W_3 + \lambda_2(V_n - V_{n-1})$$

We still have to find λ_1 and λ_2.
There are different approaches to that:

Offset Curves

Klass-Method:

The curvature radii in the end points of the original curve and the offset curve should only differ by d:

$$\frac{1}{k_i} = \frac{1}{\tilde{k}_i} - d_i; \quad i \in \{0, 1\}$$

This requirement directly yields a non-linear equation system that is solved using Newton's method.

Offset Curves

Hoschek-Method:

The basic idea of this method is to use the "degrees of freedom λ_1 and λ_2" to minimize the total error between the offset curve $X_d(t)$ and the approximation $Y(t)$ utilizing least squares fitting.

First, one computes $k+1$ points $\{P_i\}$ on the exact offset curve $X_d(t)$ and then determines the local error:
Error per interval: $\delta_i := P_i - \sum_{j=0}^{3} W_j B_j^3(t_i)$

Offset Curves

Hoschek-Method:

Total error:

$$\delta := \sum_{i=0}^{3} \delta_i^2 = \sum_{i=0}^{k} \left(D_i - \lambda_1(V_1 - V_0)B_1^3(t_i) - \lambda_2(V_n - V_{n-1})B_2^3(t_i)\right)^2$$

where D_i summarizes the known parts:

$$D_i := P_i - W_0 B_0^3(t_i) - W_1 B_1^3(t_i) - W_2 B_2^3(t_i) - W_3 B_3^3(t_i)$$

Offset Curves

Hoschek-Method:

A necessary criterion for a minimum results from:

$$\frac{\partial\delta}{\partial\lambda_1} = 0 \text{ and } \frac{\partial\delta}{\partial\lambda_2} = 0$$

This directly yields a linear equation system.

Offset Curves

Hoschek-Method:
Algorithm for Cubic Offset Curves:

step 0) choose an upper limit L for the error

step 1) compute $P_i = X_d(t_i)$, for $i = 0, \dots, k$

step 2) determine λ_1 and λ_2 by least squares fitting

step 3) determine the maximum deviation $\tilde{d} = \max \|P_i - Y(t_i)\|$

step 4) *if* $\tilde{d} < L$, stop, *else* subdivide the curve $X(t)$ and start again

Offset Curves

Hoschek-Method:

To generalize this principle, one first convinces himnself that λ_1 and λ_2 basically determine an optimal parameterization and satisfy the $\mathcal{G}^1$-conditions $X_d' = \lambda_i Y'$ between the offset curve and its approximation at the end points.

Thus, in the **quintic case**, one starts with the $\mathcal{G}^2$-conditions:

$$X_d'' = \lambda_i^2 Y'' + \mu_i Y'$$

Offset Curves

Hoschek-Method:

First, we provide the connections between the tangents and curvatures of the offset curve and the original curve. We will need them later.

Tangent of the Offset Curve Depending on the Original Curve:

$$X_d' = (1 + d \cdot \kappa_X)\, X'$$

Curvature of the Offset Curve Depending on the Curvature κ_X of the Original Curve:

$$\kappa_{X_d} = \frac{\kappa_X}{|1 + d \cdot \kappa_X|}$$

Offset Curves

Hoschek-Method:
Quintic Case:

$$W_0 := V_o + d \cdot N(a) \quad W_5 := V_n + d \cdot N(b)$$
$$W_1 := W_0 + \lambda_0(V_1 - V_0) \quad W_4 := W_5 + \lambda_2(V_n - V_{n-1})$$

$$W_2 := W_0 + \lambda_1^2 \frac{m(n-1)}{n(m-1)} \frac{1}{(1 + d \cdot \kappa_X(a))}(V_2 - V_1) + \mu_1(V_1 - V_0)$$
$$W_3 := W_5 + \lambda_2^2 \frac{m(n-1)}{n(m-1)} \frac{1}{(1 + d \cdot \kappa_X(b))}(V_{n-1} - V_{n-2}) + \mu_2(V_n - V_{n-1})$$

Offset Curves

Hoschek-Method:

Analoguosly to the cubic case, one first determines the local error δ_i and then the total error δ. The δ_i are now linear in μ_1 and μ_2 but quadratic in λ_1 and λ_2. Thus, one modifies the algorithm for cubic offset curves as follows:

Offset Curves

Hoschek-Method:
Algorithm for Quintic Offset Curves:

step 0) choose an upper limit L for the error

step 1) compute $P_i = X_d(t_i)$, for $i = 0, \ldots, k$

step 2) determine μ_1 and μ_2 by least squares fitting

step 3) utilize an optimization algorithm to minimize
$f(\lambda_1, \lambda_2) = \max \|P_i - Y(t_i, \lambda_1, \lambda_2)\|$

step 4) determine the maximum deviation $\tilde{d} = \max \|P_i - Y(t_i)\|$

step 5) *if* $\tilde{d} < L$, stop, *else* subdivide the curve $X(t)$ and start again

Offset Curves

Hoschek-Method:

Further Reading:

- One can use pretty much any optimization algorithm in step 3. A selection on optimization algorithms can be found in [Jacob, 1982]
- One can of course work with the $\mathcal{G}^3$ or $\mathcal{G}^4$ conditions and thus have more degrees of freedom available. For more information, see [Hoschek-Wissel, 1988]

Offset Curves

Hoschek-Method:

Another possibility to provide additional parameters is the approximation of the offset curve by rational Bézier curves:

$$Y(t) = \frac{\sum_{i=0}^{m} \beta_i W_i B_i^m(t)}{\sum_{i=0}^{m} \beta_i B_i^m(t)}$$

The $\mathcal{G}^1$-conditions now yield:

$$W_0 := V_0 + d \cdot N(a) \quad W_3 := V_n + d \cdot N(b)$$

$$W_1 := W_0 + \lambda_1 \frac{\beta_0}{\beta_1}(V_1 - V_0) \quad W_2 := W_3 + \lambda_2 \frac{\beta_3}{\beta_2}(V_n - V_{n-1})$$

Note that this can be extended to the quintic case utilizing the $\mathcal{G}^2$-conditions analoguosly to before.

Offset Curves

Hoschek-Method:

W.l.o.g., we now set $\beta_0 = 1 = \beta_3$. After choosing $2n+1$ points P_i on the offset curve $X_d(t)$, one can again determine an (again, at first local) error function δ_i:

$$\delta_i = (P_i - W_0)B_0^3(t_i) - \beta_1(W_0 - P_i)B_1^3(t_i) - \beta_2(W_3 - P_i)B_2^3(t_i) + (P_i - W_3)B_3^3(t_i) - \lambda_1(V_1 - V_0)B_1^3(t_i) - \lambda_2(V_n - V_{n-1})B_2^3(t_i)$$

Offset Curves

Hoschek-Method:

The error functions δ_i are linear in β_1, β_2, λ_1, and λ_2.
Thus, the total error

$$\delta = \sum_{i=0}^{2n+1} \delta_i^2$$

can be determined by least squares fitting.

Offset Curves

Hoschek-Method:
Algorithm for Rational Cubic Offset Curves:

step 0) choose an upper limit L for the error

step 1) compute $P_i = X_d(t_i)$, for $i = 0, \ldots, 2n+1$

step 2) determine β_1, β_2, λ_1, and λ_2 by least squares fitting (remember: β_0 and β_3 have been set to 1)

step 3) determine the maximum deviation $\tilde{d} = \max \|P_i - Y(t_i)\|$

step 4) *if* $\tilde{d} < L$, stop, *else* subdivide the curve $X(t)$ and start again

Offset Curves

The methods of Klass and Hoschek are efficient further developments of a basic idea by Blomgren (see [Tiller-Hanson, 1984]):

1. offset each line segment of the original curve
2. check the maximum deviation
3. subdivide until deviation suitably small

Offset Curves

Fitting-Method:

The offset curve algorithms can of course also be formulated without optimization parameters (see [Tiller-Hanson, 1984], [Pham, 1988]):

step 1) compute the points of the exact offset curve

step 2) interpolate these points (perhaps also the respective tangents) with a rational or polynomial B-spline curve or approximate using least squares fitting

step 3) check the deviation $\tilde{\delta}$

step 4) *if* $\tilde{\delta} < L$, stop, *else* subdivide the original curve and start again

Offset Curves

Problems with Peaks and Self-Intersections:
The investigation of offset curves is not at all throughly done by providing efficient algorithms. Even a "reasonable" original curve $X(t)$ (i.e. a curve without cusps and self-intersections) does not guarantee a "feasible" offset curve as the following example demonstrates:

Offset Curves

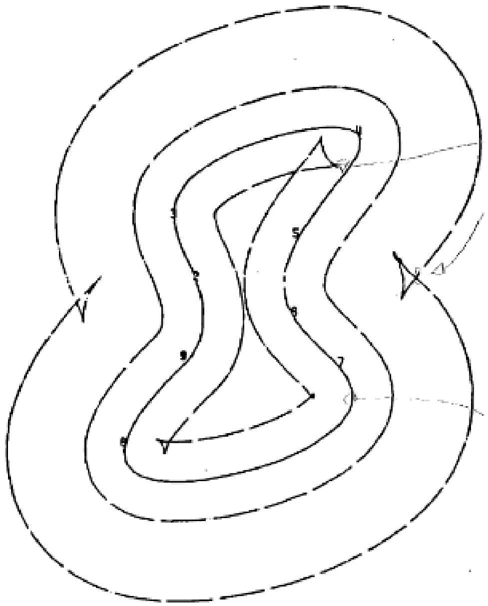


Figure: Example where a original curve without cusps and self-intersections (the second from the inner) yields to cusps and self-indersections for inner and out offset curves.

One thus also has to find requirements for the original curve and for the distance d that guarantee offset curves without cusps and self-intersections.

Offset Curves

The connections between the offset curve's (X_d) and the original curve's (X) tangent and curvature

$$X_d' = (1 + d \cdot \kappa_X) \cdot X' \text{ and } \kappa_{X_d} = \frac{\kappa_X}{1 + d \cdot \kappa_X}$$

directly yield the **"regularity criterion"** $1 + d\kappa_X(t) \neq 0$

Offset Curves

However, one needs the respective parameters t_i, where the zero-crossing are. These can only be computed very expensively and extremely error-prone numerically from

$$f(t) = 1 + d \cdot \kappa_X(t) = 0$$

Offset Curves

A substantially more effective way is to interpret the curve as an envelope (dt. Einhüllende) of its tangents and to compute the cusps of the offset curve (if existent) from a determinant-based criterion (see [Hoschek, 1985]).

Every tangent of a planar (torsion-free) curve can be represented by:

$$V_1\xi + V_2\eta + V_0 = 0$$

where $V_0 := \dot{X}Y - X\dot{Y}$, $V_1 := -\dot{Y}$, and $V_2 := \dot{X}$, provided the curve is parameterized by $X(t) = (X(t), Y(t))^T$.

Offset Curves

The curve $X(t)$ then has a cusp if and only if the function $g(t) = \det(V, \dot{V}, \ddot{V})$ crosses zero and (!) changes its sign at this crossing ($V := (V_0, V_1, V_2)^T$). If there is no sign change, the zero-crossing is just a point of inflection (dt. Wendepunkt).

These geometric facts together with of Newton's method for the approximation of the zeros of a real-valued function yield an efficient algorithm for the determination and removal of cusps from offset curves (see [Hoschek, 1985])

Offset Curves

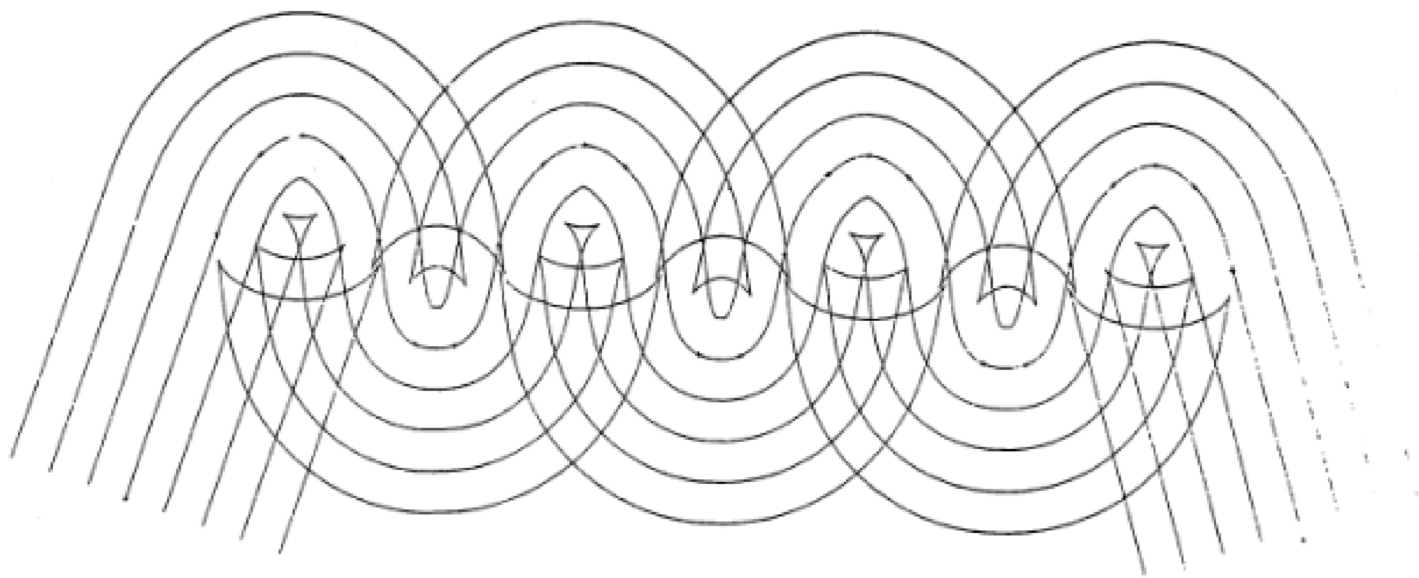
Algorithm for Regularization of Offset Curves:

If $f(t) = 1 + d \cdot \kappa_X(t)$, the curve is regular. *Else*:

1. determine the zero-crossing of $\det(V, \dot{V}, \ddot{V})$
2. remove loops
3. if desired, smoothen the cusps

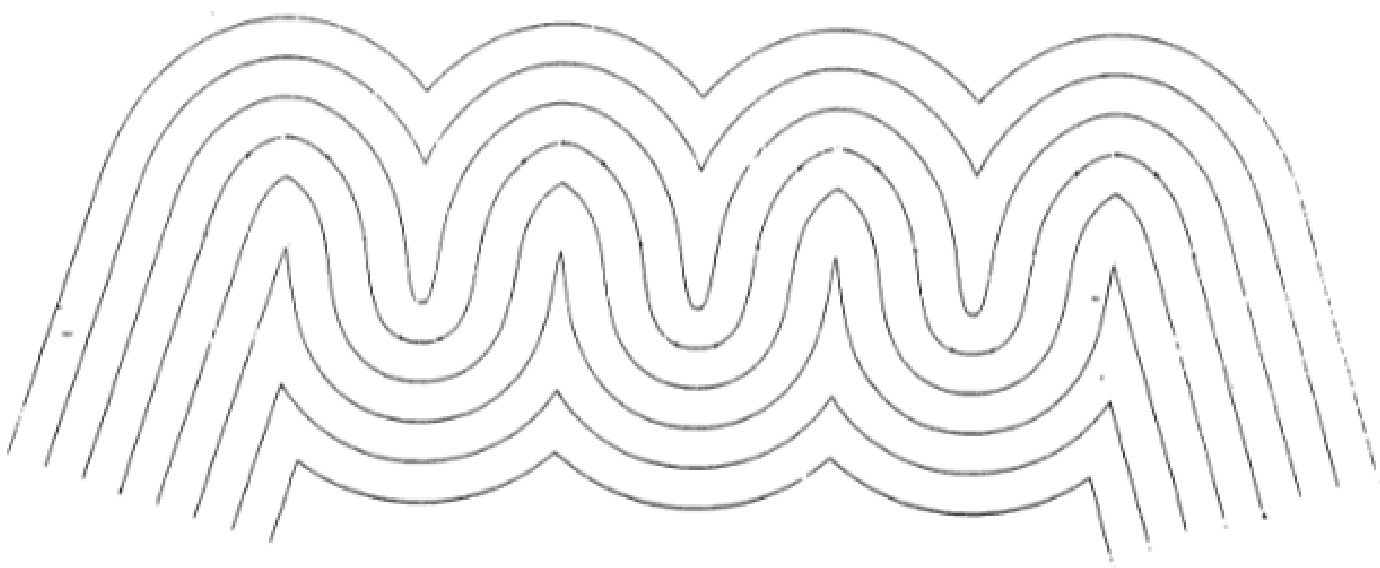
Offset Curves

Before we proceed to discuss the step "remove loops", we provide an application example where the smoothing of cusps makes sense to construct a desired pattern (see [Hoschek, 1985]):



Offset Curves

The removal of loops and the smoothing of the resulting cusps then yields the desired pattern:

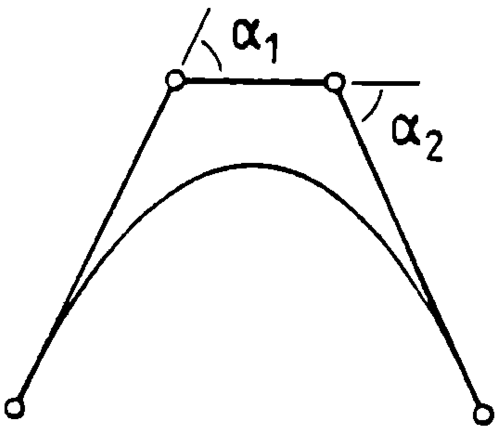


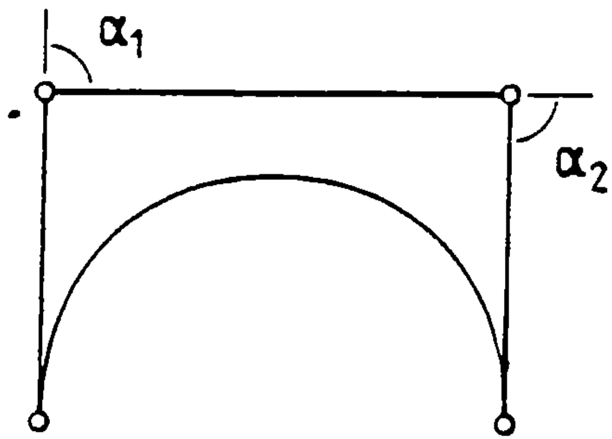
The central problem of the "remove loops" step is the computation of the self-intersections.

Offset Curves

Lasser-Method (see [Lasser, 1989]):

The essential basic idea of this algorithm is the following angle criterion:
A Bézier curve does not have self-intersections if the sum of all absolute values of the outer rotation angles of the Bézier polygon is smaller than or equal to π, i.e. $\sum |\alpha_k| \leq \pi$.

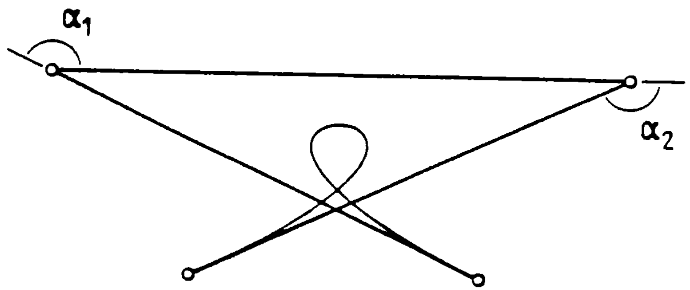


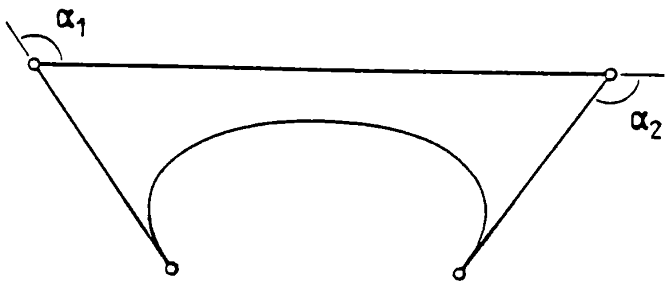


Offset Curves

Lasser-Method (see [Lasser, 1989]):

If $\sum |\alpha_k| \leq \pi$, we know that the curve does not intersect itself. For $\sum |\alpha_k| > \pi$ instead, self-intersections may occur:





For more detail concerning this universal and stable method, see [Lasser, 1989].