Algorithmic Geometry WS 2017/2018

Prof. Dr. Hans Hagen
Benjamin Karer M.Sc.

http://gfx.uni-kl.de/~alggeom

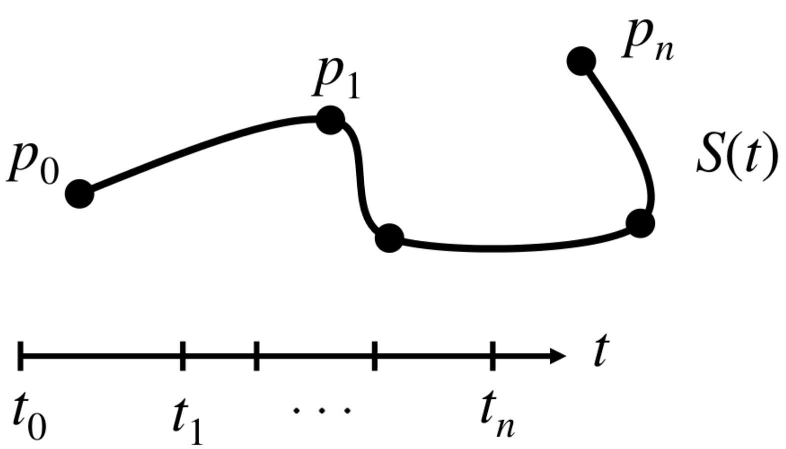
Spline Curves

Spline Curves

Problem: In the previous chapter, we have seen that interpolating polynomials, especially those of high degree, tend to produce strong wriggling effects.

Solution: Use curves consisting of several low-degree segments.

Condition: The polynomial segments have to fit together "smoothly" at the transitions (nodes).



$S(t_i) = p_i$

Definition

C^k-continuity

- A function $f(t)$ is C^k-continuous, if the function and its first k derivatives are continuous.
- $C^k[t_0, t_n]$ is the class of C^k-continuous functions on the interval $[t_0, t_n]$.

Definition

Spline

- Let $\tau = \{t_0, \ldots, t_n\}$ a node vector with real-valued nodes $t_i < t_{i+1}$.
- A function S is called *Spline* of degree k (of order $k+1$) if:
  1. S is a polynomial of degree k in every partial interval $[t_i, t_{i+1}]$
  2. S is C^{k-1}-continuous on $[t_0, t_n]$.

Remarks

1. The spline S is called *interpolating spline*, if $S(t_i) = p_i$ for a given set of interpolation points p_i (ordinates).
2. In general, the interpolating spline is not uniquely defined. Therefore, additional boundary conditions are required for the remaining $k-1$ degrees of freedom.

Natural Splines: cubic splines ($k = 3$) with natural boundary conditions $S''(t_0) = 0$ and $S''(t_n) = 0$

Periodic Splines: identify nodes t_0 and t_n with each other, i.e.

$$S(t_0) = S(t_n);\ S'(t_0) = S'(t_n);\ S''(t_0) = S''(t_n)$$

Cubic Splines

Instead of specifying properties of the interpolating function (e.g. maximal degree), we can also specify the shape (e.g. "smooth").
Requirement:

$$\int_{t_0}^{t_n} \left\| g''(t) \right\|^2 \quad \rightarrow \quad \text{minimal} \tag{1}$$

with additional conditions:

$$g(t_j) = p_j \quad (j = 0, \ldots, n), \quad g'(t_0) = p'_0 \quad \text{and} \quad g'(t_n) = p'_n \tag{2}$$

Theorem

Minimum-Norm Property
Among all functions $g \in C^2[t_0, t_n]$ which satisfy $g(t_i) = y_i$, the integral $\int_{t_0}^{t_n} \|g''(x)\|^2$ for the interpolating cubic spline function S has the smallest value.

Proof: ...

Interpolation with Natural Cubic Splines

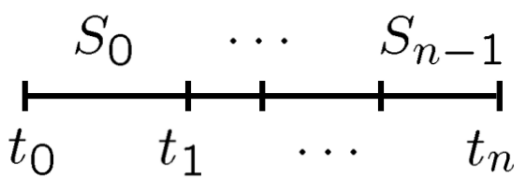
Coefficients a_i, b_i, c_i, d_i of a natural cubic spline:

$$S(t) = S_i(t) = a_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3$$

for $t \in [t_i, t_{i+1}]$, $i = 0, \dots, n-1$.

$\rightarrow$ Conditions for polynomials S_i:

$$\begin{aligned} S_i(t_i) &= p_i, & i &= 0, \dots, n-1, \\ S_i(t_{i+1}) &= p_{i+1}, & i &= 0, \dots, n-1, \\ S_i'(t_i) &= S_{i-1}'(t_i), & i &= 1, \dots, n-1, \\ S_i''(t_i) &= S_{i-1}''(t_i), & i &= 1, \dots, n-1. \end{aligned}$$



For our coefficients, this implies that:

$$
\begin{aligned}
a_i &= p_i, \quad i = 0, \ldots, n-1 \text{ and } S_{n-1}(t_n) = p_n, \\
a_i &= a_{i-1} + b_{i-1}(t_i - t_{i-1}) + c_{i-1}(t_i - t_{i-1})^2 \\
&\quad + d_{i-1}(t_i - t_{i-1})^3, && i = 1, \ldots, n-1 \\
b_i &= b_{i-1} + 2c_{i-1}(t_i - t_{i-1}) + 3d_{i-1}(t_i - t_{i-1})^2, && i = 1, \ldots, n-1 \\
2c_i &= 2c_{i-1} + 6d_{i-1}(t_i - t_{i-1}), && i = 1, \ldots, n-1
\end{aligned}
$$

We define $\Delta_i := t_{i+1} - t_i$ and, after some transformations, get:

$$c_{i-1}(\Delta_{i-1}) + c_i(2(\Delta_{i-1} + \Delta_i)) + c_{i+1}(\Delta_i) = \frac{3}{\Delta_i}(p_{i+1} - p_i) - \frac{3}{\Delta_{i-1}}(p_i - p_{i-1}) \qquad i = 1, \dots, n-1 \quad (3)$$

with $c_n := 0$

$$d_i = \frac{1}{3\Delta_i}(c_{i+1} - c_i) \qquad i = 0, \dots, n-1 \quad (4)$$

$$b_i = \frac{1}{\Delta_i}(p_{i+1} - p_i) - \frac{\Delta_i}{3}(c_{i+1} + 2c_i) \qquad i = 0, \dots, n-1 \quad (5)$$

$a_i = p_i, \quad i = 0, \ldots, n-1, \quad$ i.e. the a_i are already known.
$\rightarrow$ We have to solve a linear system of $n-1$ equations with $n+1$ unknowns (c_i, $i = 0, \ldots, n$).

Natural Cubic Splines

$$S''(t_0) = 0 \text{ and } S''(t_n) = 0$$
$$\rightarrow c_0 = c_n = 0$$

We have:
- $n+1$ supporting values (nodes) t_i with $t_0 < t_1 < \cdots < t_n$ and
- function values $p_0, \ldots, p_n$.

We want: natural cubic spline S of the form

$$S(t) = S_i(t) = a_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3$$

for $t \in [t_i, t_{i+1}]$ and $i = 0, \ldots, n-1$.

The coefficients a_i, b_i, c_i, d_i are found using (3)-(5) and

$$a_i = p_i \quad i = 0, \ldots, n-1$$
$$c_0 = c_n = 0$$

The equations (3) can be written in matrix form $Ac = r$:

$$A := \begin{pmatrix} 2(\Delta_0 + \Delta_1) & \Delta_1 & 0 & . & . \\ \Delta_1 & 2(\Delta_1 + \Delta_2) & \Delta_2 & . & 0 \\ 0 & \Delta_2 & . & . & \Delta_{n-2} \\ . & . & 0 & \Delta_{n-2} & 2(\Delta_{n-2} + \Delta_{n-1}) \end{pmatrix}$$

$$c := \begin{pmatrix} c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} \qquad r := 3 \begin{pmatrix} \frac{p_2 - p_1}{\Delta_1} - \frac{p_1 - p_0}{\Delta_0} \\ \vdots \\ \frac{p_n - p_{n-1}}{\Delta_{n-1}} - \frac{p_{n-1} - p_{n-2}}{\Delta_{n-2}} \end{pmatrix}$$

The matrix A is tridiagonal, symmetric, diagonally dominant, positive-definite, and consists of only positive elements. This implies:

- A is regular and the linear system of equations has a unique solution.
- For solving the system, the direkt LU-decomposition for tridiagonal matrices should be used, because the algorithm has the complexity of only $\mathcal{O}(n)$.

Interpolation with Periodic Cubic Splines

If we identify t_0 with t_n, i.e. $p_0 = p_n$, we get a closed interpolating curve with a C^2-continuous transition at t_0.
$\rightarrow$ In the algorithm for natural cubic splines, we only have to change the matrix A of system (3):

$$A := \begin{pmatrix} 2(\Delta_0 + \Delta_1) & \Delta_1 & . & 0 & \Delta_0 \\ \Delta_1 & 2(\Delta_1 + \Delta_2) & \Delta_2 & . & 0 \\ 0 & . & . & . & \Delta_{n-1} \\ \Delta_0 & 0 & . & \Delta_{n-1} & 2(\Delta_{n-1} + \Delta_0) \end{pmatrix}$$

This matrix is cyclic-tridiagonal, symmetric, diagonally dominant, positive-definite and consists of only positive elements. This implies:

- A is well-conditioned
- The system can also be solved in $\mathcal{O}(n)$.
- In comparison to the system (3) for natural cubic splines, the matrix is larger by one row and one column, because we have n instead of $n-1$ transitions.
- Due to the cyclic band structure of the matrix, we have additional non-zero elements in the upper right and lower left corners.

Polynomial Averaging Splines

We have:
- Values $f(t_i) = f_i$ of a function $f \in C[a,b]$ with errors (e.g. measurement errors)
- $n+1$ nodes t_i with $a = t_0 < t_1 < \cdots < t_n = b$

- The distribution of the f_i makes a useful approximation with interpolating splines impossible.
- We need an "error-compensating replacement function", which runs "smoothly" along the points (t_i, f_i).

Consider an interpolating spline S through a new set of values g_i. The g_i have to satisfy that the differences $f_i - g_i$ are positively proportional to the jumps γ_i of the third derivative of S at t_i.

Definition

Polynomial Averaging Spline

A polynomial averaging spline of degree three with $n+1$ nodes t_i and "flawed" values f_i is a function $S : [a, b] \to \mathbb{R}$ with the following properties:

$$S \in C^2[a, b] \tag{6}$$

$$S \text{ is a polynomial of degree three in each sub-interval} \tag{7}$$

$$S(t_i) = g_i \quad \text{for } i = 0, \ldots, n \tag{8}$$

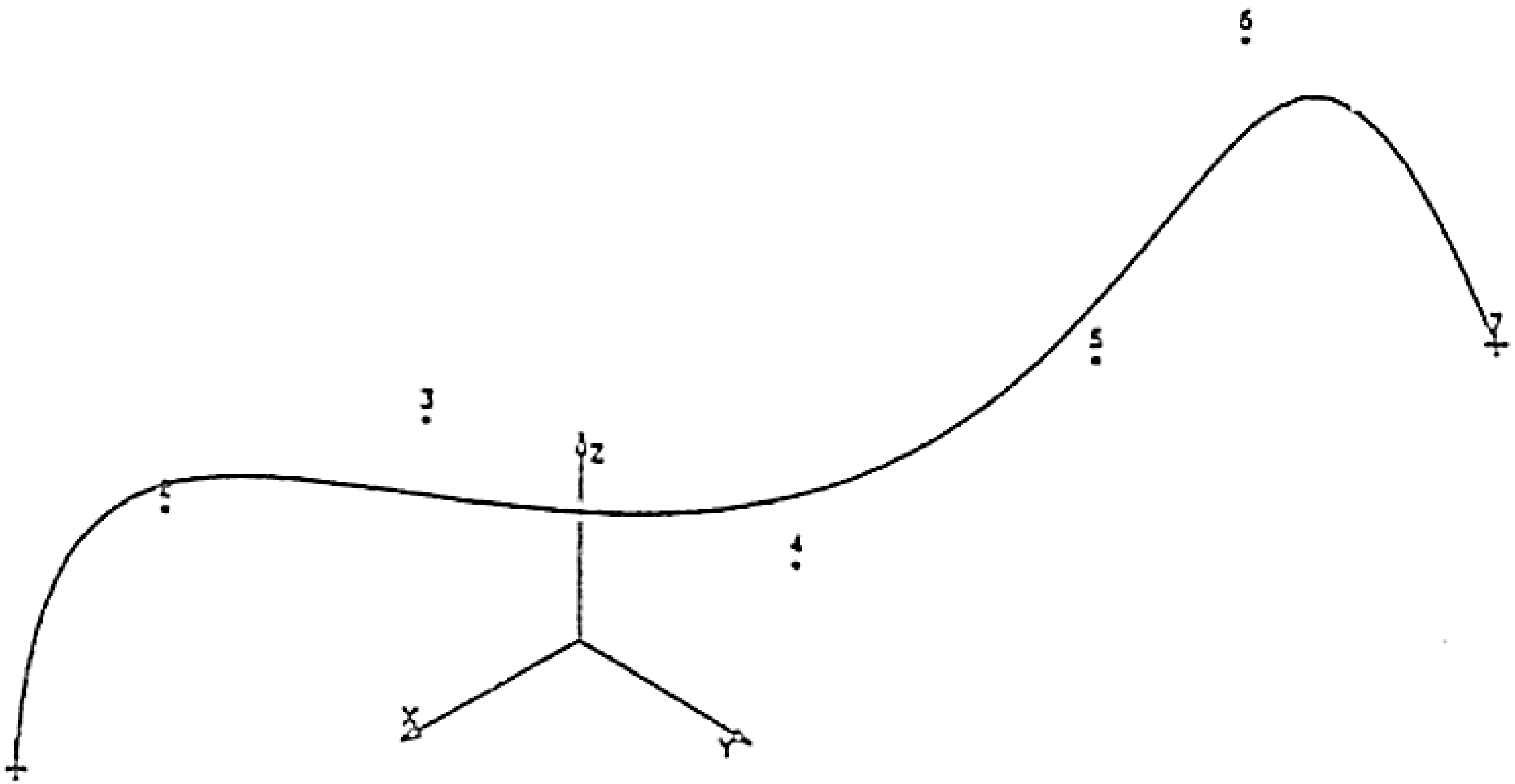
$$w_i \cdot (f_i - g_i) = \gamma_i \quad \text{for } i = 0, \ldots, n \text{ with weights } w_i > 0 \tag{9}$$

$$\gamma_0 = S_0'''(t_0) \quad \gamma_n = -S_{n-1}'''(t_n)$$

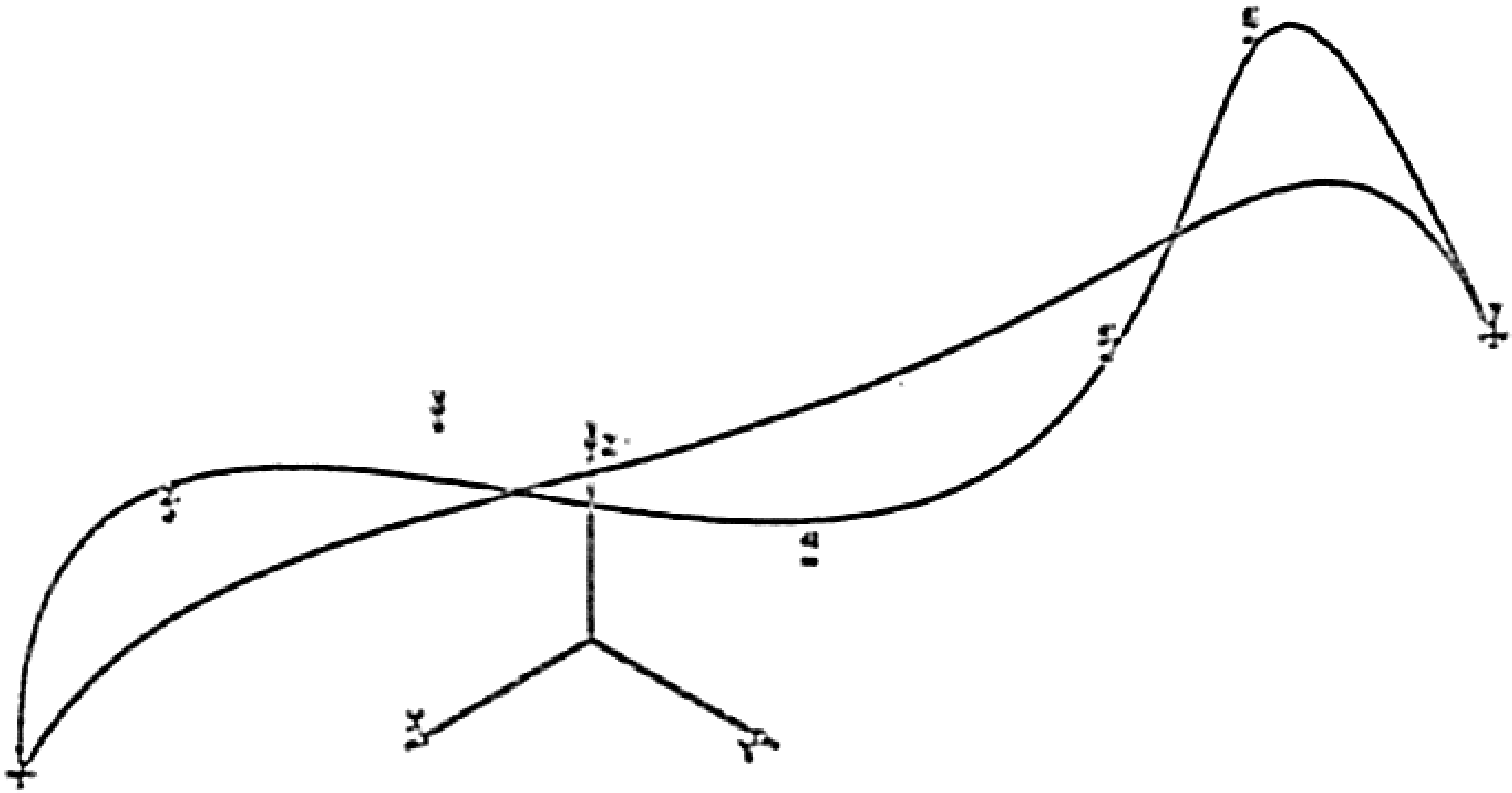
$$\gamma_i = S_i'''(t_i) - S_{i-1}'''(t_i) \quad i = 1, \ldots, n-1$$

Spline Curves

Choosing large values for the weights w_i will result in a curve that is very close to the data values f_i, while smaller w_i will result in a flatter curve.



Spline Curves



The system of equations used by an averaging spline algorithm can be constructed by combining the conditions for interpolating splines with the specific conditions for averaging splines (9).

Parametric Splines

Modelling of plane curves or space curves:
$\rightarrow$ *vector-valued* or *parametric* splines

Definition

Parametric Spline
Consider an interval $[a, b] \subset \mathbb{R}$ and $\Delta := (t_0, \ldots, t_n)$, $a = t_0 < t_1 < \cdots < t_n = b$. The mapping $X : [a, b] \rightarrow \mathbb{R}^3$ is called *parametric spline* of degree k (order $k + 1$), if its component functions x_i, $i = 1, 2, 3$ are of degree k:

$$x_i \in C^{k-1}[a, b] \; (i = 1, 2, 3)$$

$$\text{short: } X \in C^{k-1}[a, b]$$

Definition

C^k-Transition

Two parametric curves $X : [t_0, t_1] \to \mathbb{R}^3$ and $Y : [s_0, s_1] \to \mathbb{R}^3$ with $X \in C^m[t_0, t_1]$ and $Y \in C^n[s_0, s_1]$ have a C^k*-transition* at their shared point $X(t_1) = Y(s_0)$ if the following holds:

$$\frac{d^r}{dt^r} X(t_1) = \frac{d^r}{ds^r} Y(s_0)$$

for all r with $1 \leq r \leq k$.

Discontinuities are also called C^{-1}-transitions.

Interpolation with Parametric Cubic Splines

We have: interpolation points $p_i := (x_i, y_i, z_i), \; i = 0, \ldots, n$.

We want: interpolating parametric cubic spline $S(t)$.

Step 1: Parametrization

Specify parameter values (nodes) $t_i, i = 0, \ldots, n$ for the interpolation points: $S(t_i) = p_i$.

Step 2: Boundary Conditions

Specify boundary conditions (natural or periodic).

Step 3: Spline algorithm

Calculate the spline components S_x, S_y, S_z such that $S_x(t_i) = x_i, S_y(t_i) = y_i, S_z(t_i) = z_i,\ \ i = 0, \dots, n$ using the appropriate spline algorithm:

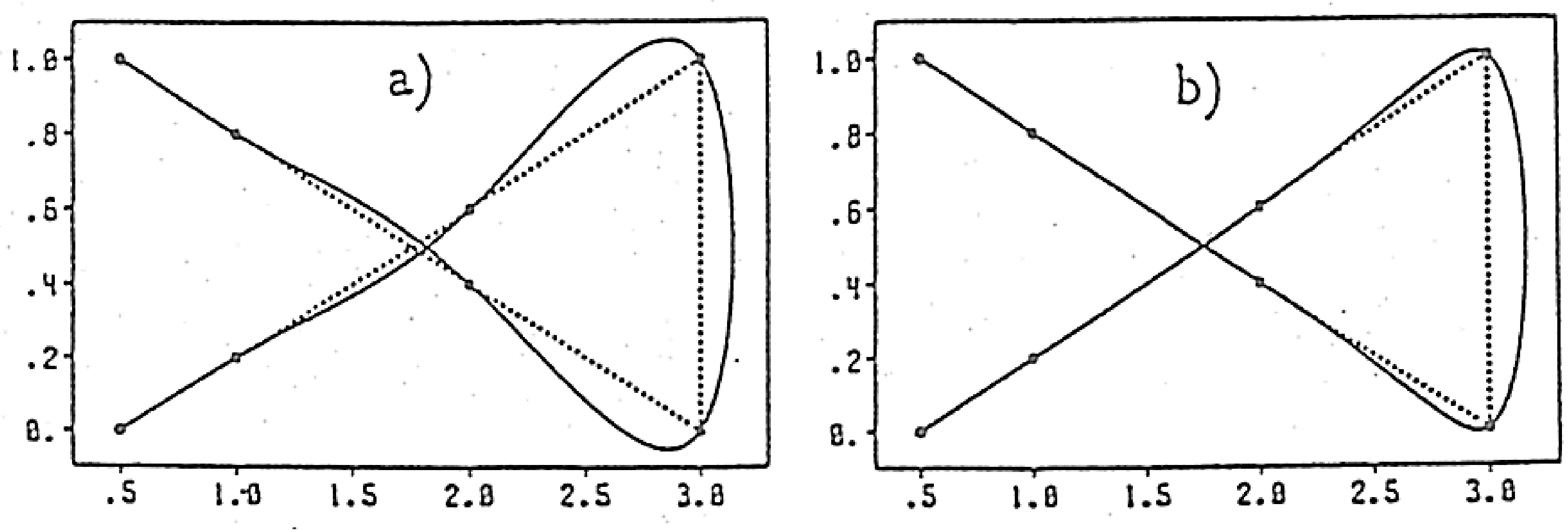
$$
\begin{aligned}
S_x(t) &= S_{xi}(t) = a_{xi} + b_{xi}(t - t_i) + c_{xi}(t - t_i)^2 + d_{xi}(t - t_i)^3 \\
S_y(t) &= S_{yi}(t) = a_{yi} + b_{yi}(t - t_i) + c_{yi}(t - t_i)^2 + d_{yi}(t - t_i)^3 \\
S_z(t) &= S_{zi}(t) = a_{zi} + b_{zi}(t - t_i) + c_{zi}(t - t_i)^2 + d_{zi}(t - t_i)^3 \\
&\quad t \in [t_i, t_{i+1}],\ \ i = 0, \dots, n - 1.
\end{aligned}
$$

Spline Curves

- For closed curves which are “smooth” everywhere, it makes sense to use periodic splines.
- If a curve has one or more “cusps” (i.e. C^0-transitions), we can use natural splines with the cusps as start- and endpoints.

Parametrizations

The shape, and therefore also the quality, of a curve (or surface) depends strongly on the parametrization.



(a) and (b) are two different parametrizations for the same set of interpolation points.

We can demonstrate the impact of the choice of parameters using a kinematic interpretation:

- Interpret the curve parameter t as a time parameter.
- It represents the amount of time that a point S would need to traverse the curve.

For the following parametrizations, we are interpolating a set of points using a curve:

- parameter interval $[a, b]$ and
- $n + 1$ points to interpolate

Equidistant Parametrization

For each pair of successive interpolations points (p_i, p_{i+1}), we have the same amount of time for traversal:

$$\Delta t = \frac{b-a}{n};\;\; t_i = a + i \cdot \Delta t;\;\; i = 0, \ldots, n$$

If the distances between the points vary strongly, then a point S traverses the curve with varying speed. That is, if a large distance is followed by a small distance, the speed has to be reduced greatly. This can result in “wiggling” of the interpolation curve.

Chordal Parametrization

Idea: Adapt the parametrization to the "structure" of the point set. This is achieved by selecting the parameter intervals proportionally to the distances of neighboring interpolation points. We control the overall length using a normalization factor s (example: s is the overall length of the polygon formed by the p_i).

$$\Delta t_i = t_{i+1} - t_i := \frac{\|p_{i+1} - p_i\|}{s}.$$

Centripetal Parametrization

Idea: The parametrization should minimize the centripetal acceleration (normal acceleration) [Lee 1989]. The normal forces along an arc are proportional to the angular velocity.

$$\Delta t_i := \frac{\sqrt{\|p_{i+1} - p_i\|}}{s}$$

These parametrizations are not affine invariant because length measurements are used. For example, the expression to the right is affine invariant only if the three points lie in a line.

$$\frac{\|p_{i+1} - p_i\|}{\|p_{i+2} - p_{i+1}\|}$$

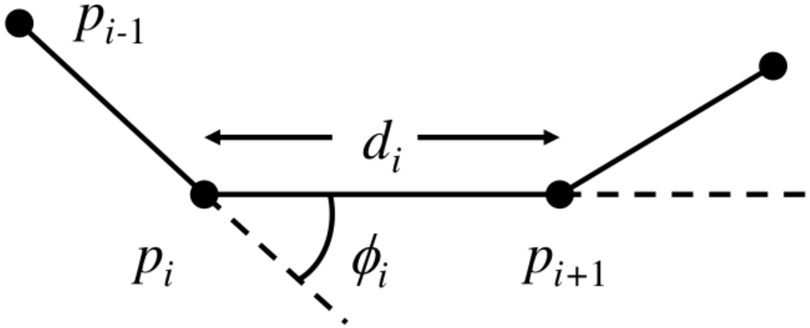
Foley-Parametrization

Idea: The parametrization should take into account distances and angle differences at the interpolation points [Foley 1989].

$$\Delta t_i := d_i \left(1 + \frac{2 \cdot \tilde{\phi}_i \cdot d_{i-1}}{3(d_{i-1} + d_i)} + \frac{2 \cdot \tilde{\phi}_{i+1} \cdot d_{i+1}}{3(d_i + d_{i+1})} \right)$$

with

$$\tilde{\phi}_i := \min(\pi - \phi_i, \frac{\pi}{2}).$$



Nielson-Metric

The distance function d used with the Foley-parametrization can be either the euclidean metric or the Nielson-metric, which is affine-invariant:

$$\left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_N^2 := (x, y) \cdot \begin{pmatrix} \frac{\sigma_y}{\Delta} & \frac{-\sigma_{xy}}{\Delta} \\ \frac{-\sigma_{xy}}{\Delta} & \frac{\sigma_x}{\Delta} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\overline{x} := \frac{1}{n} \sum_{i=1}^{n} x_i; \qquad \overline{y} := \frac{1}{n} \sum_{i=1}^{n} y_i;$$

$$\text{with } \sigma_x := \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2; \quad \sigma_y := \frac{1}{n} \sum_{i=1}^{n} (y_i - \overline{y})^2;$$

$$\sigma_{xy} := \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x}) \cdot (y_i - \overline{y}); \quad \Delta := \sigma_x \sigma_y - \sigma_{xy}^2$$

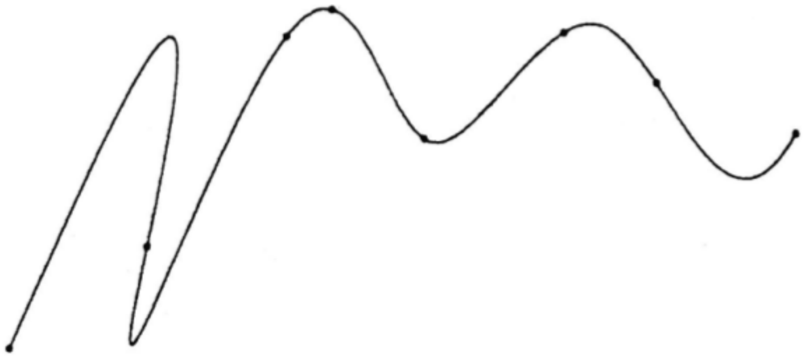
The Nielson-metric works as follows:

- scale the point set in a way that the variance is equal in all directions
- distances are obtained from the scaled arrangement of the points

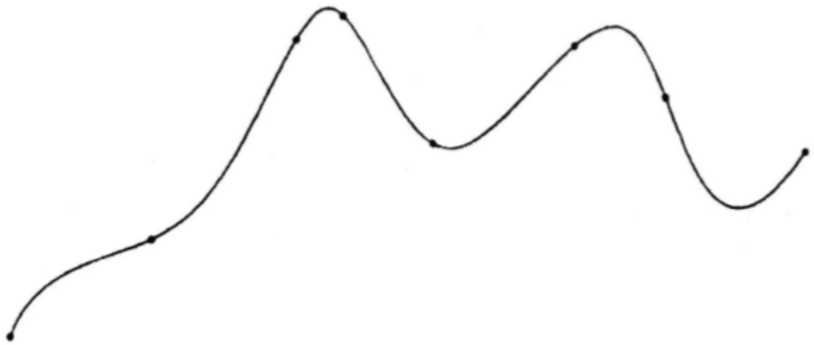
The metric allows for independence from coordinate systems and scaling.

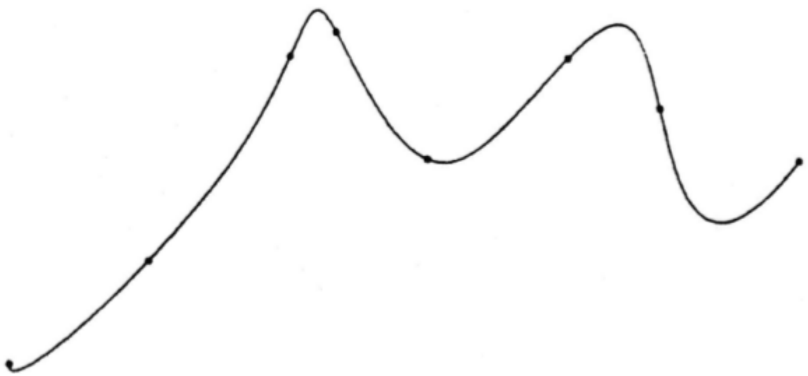
Example

equidistant parametrization

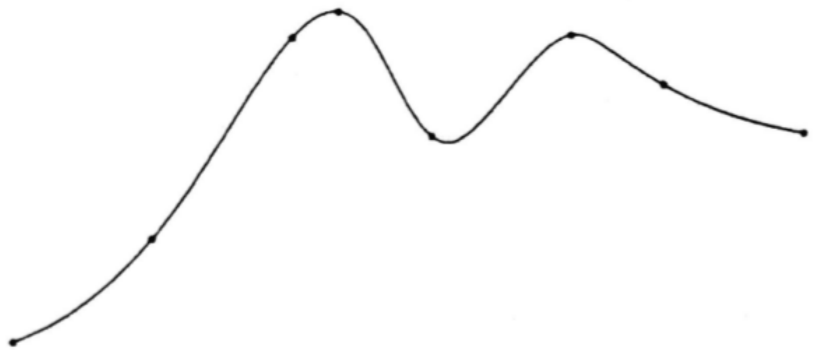


centripetal parametrization





chordal parametrization



Foley-parametrization with
Nielson metric

Parameter Transformations

The shape of a curve is strongly influenced by the parametrization of the interpolation points. It is, however, possible to modify the parametrization without altering the shape of the curve.

Definition

Parameter Transformation, Reparametrization
Consider a parametric curve $X(t)$ and a bijective, continuous function $\varphi(t)$. Then the curve $Y(t) : X(\varphi(t))$ can be obtained from X using a *parameter transformation*.
If both φ and φ^{-1} are continuously differentiable, φ is called a C^1 parameter transformation.

Example

The length of a curve segment is

$$L(t_0, t_1) = \int_{t_0}^{t_1} \left\| \dot{X}(t) \right\| dt, \quad \dot{X} = \frac{dX}{dt}$$

Using this, we can, for example, re-parametrize the curve such that $\left\| \dot{X}(t) \right\| = 1$ and t is the arc length of the curve.

The following properties do **not** depend on the parametrization of the curve:

- Curvature ("Deviation with respect to the tangent") and
- Torsion ("Deviation from planar shape")

Definition

$$\varkappa(t) = \frac{\left\| \dot{X}(t) \times \ddot{X}(t) \right\|}{\left\| \dot{X}(t) \right\|^3} \qquad \text{(curvature)}$$

$$\tau(t) = \frac{\det\left(\dot{X}(t), \ddot{X}(t), \dddot{X}(t)\right)}{\left\| \dot{X}(t) \times \ddot{X}(t) \right\|^2} \qquad \text{(torsion)}$$

The radius of the *osculating circle* (German: Schmiegekreis) of $X(t)$ is $r(t) = \frac{1}{\varkappa(t)}$.

Example

Proof that the curvature of $Y(t) = X(\varphi(t))$ does not depend on φ:

$$\dot{Y}(t) = \frac{dY}{dt} = X'(\varphi(t))\varphi'(t) =: X'\varphi'$$

$$\ddot{Y}(t) = \frac{d\dot{Y}}{dt} = X''(\varphi(t))(\varphi'(t))^2 + X'(\varphi(t))\varphi''(t) =: X''\varphi'^2 + X'\varphi''$$

$$\varkappa_Y(t) = \frac{\left\|(X'\varphi') \times (X''\varphi'^2 + X'\varphi'')\right\|}{\|X'\varphi'\|^3}$$

$$= \frac{\varphi'^3 \|X' \times X''\| + \varphi'\varphi'' \overbrace{\|X' \times X'\|}^{0}}{\varphi'^3 \|X'\|^3}$$

$$= \frac{\|X' \times X''\|}{\|X'\|^3} = \varkappa_X(\varphi(t))$$

Remark

Instead of C^k-continuity, continuity of geometric properties (e.g. tangent, curvature, torsion,...) may be required.

$\rightarrow$ As a result, we have more degrees of freedom for modelling.
$\rightarrow$ *geometric splines, G-splines*
$\rightarrow$ see lectures on

Geometric Modelling