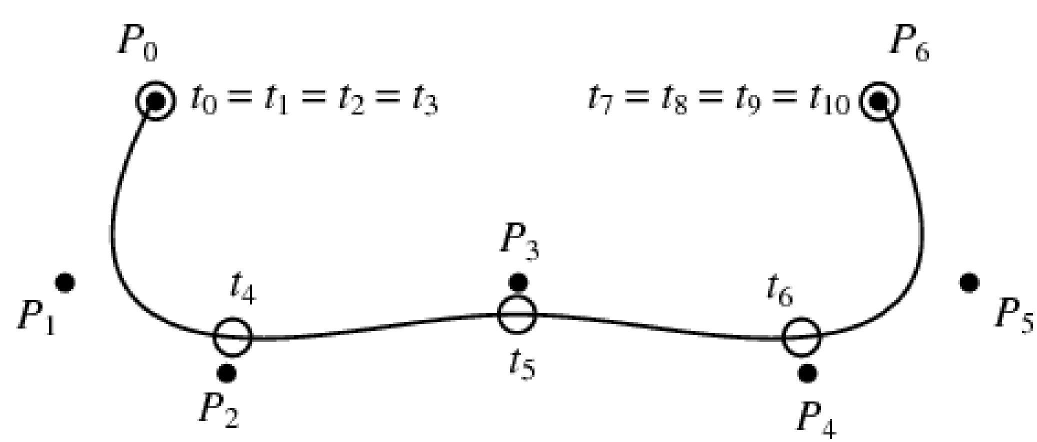
Outline

- Interpolation and polynomial approximation
- Interpolation
  - Lagrange
  - Cubic Splines

Approximation

  - Bézier curves
  - B-Splines

- Some vocabulary (again ;)
  - Control point : Geometric point that serves as support to the curve
  - Knot : a specific value of the parameter u corresponding to a joint between pieces of a curve
  - Knot sequence : the set of knots values (in an increasing order).



Wolfram research

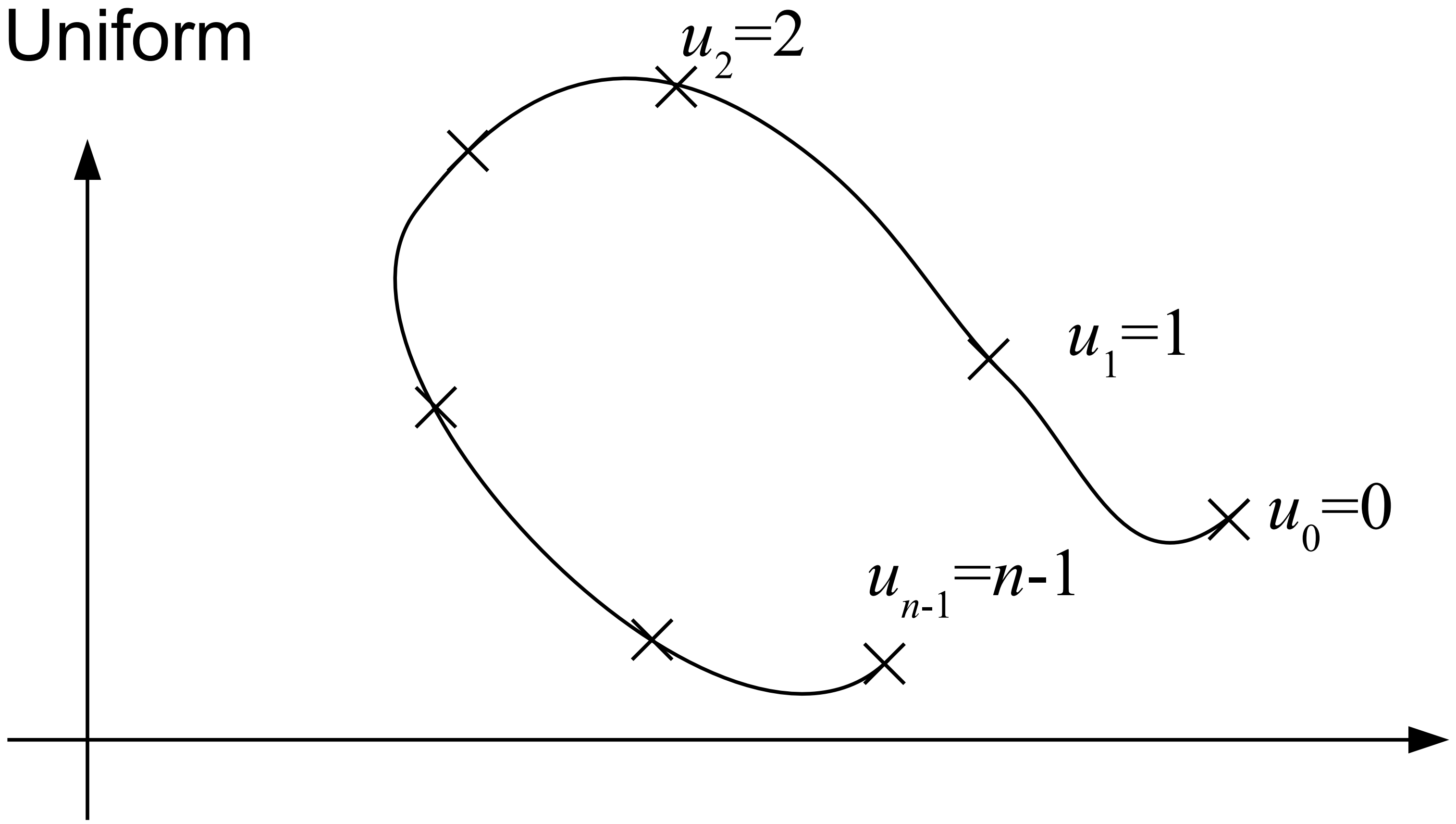
- Interpolation
  - The curve passes through the control points
- Approximation
  - The curve doesn't necessary passes through the control points
    - But these have an influence ...
  - Statistic approaches ?
    - Least squares
    - « Kriging »
    - Not very adapted to geometric modelling

Interpolation

Interpolation

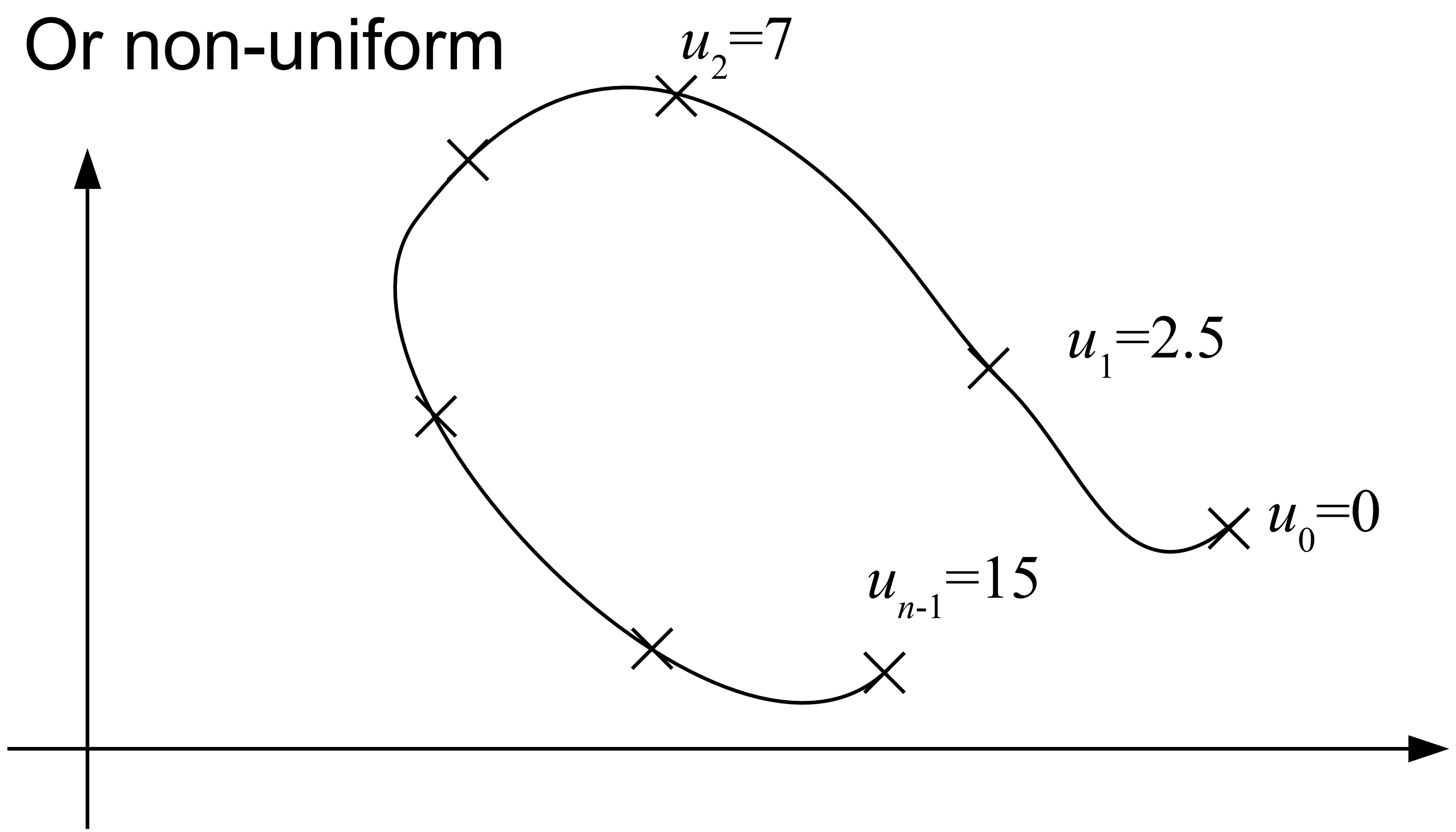
- We want draw a regular and smooth parametric curve through a certain number n of points P_i
  - Several families of base functions are available
  - Most obvious are polynomials
  - There are others -
    - Trigonometric functions (by mean of a Fourier decomposition for instance)
    - Power functions
    - etc...

- We must choose the parametrization (nodal sequence)
  - Uniform



Interpolation

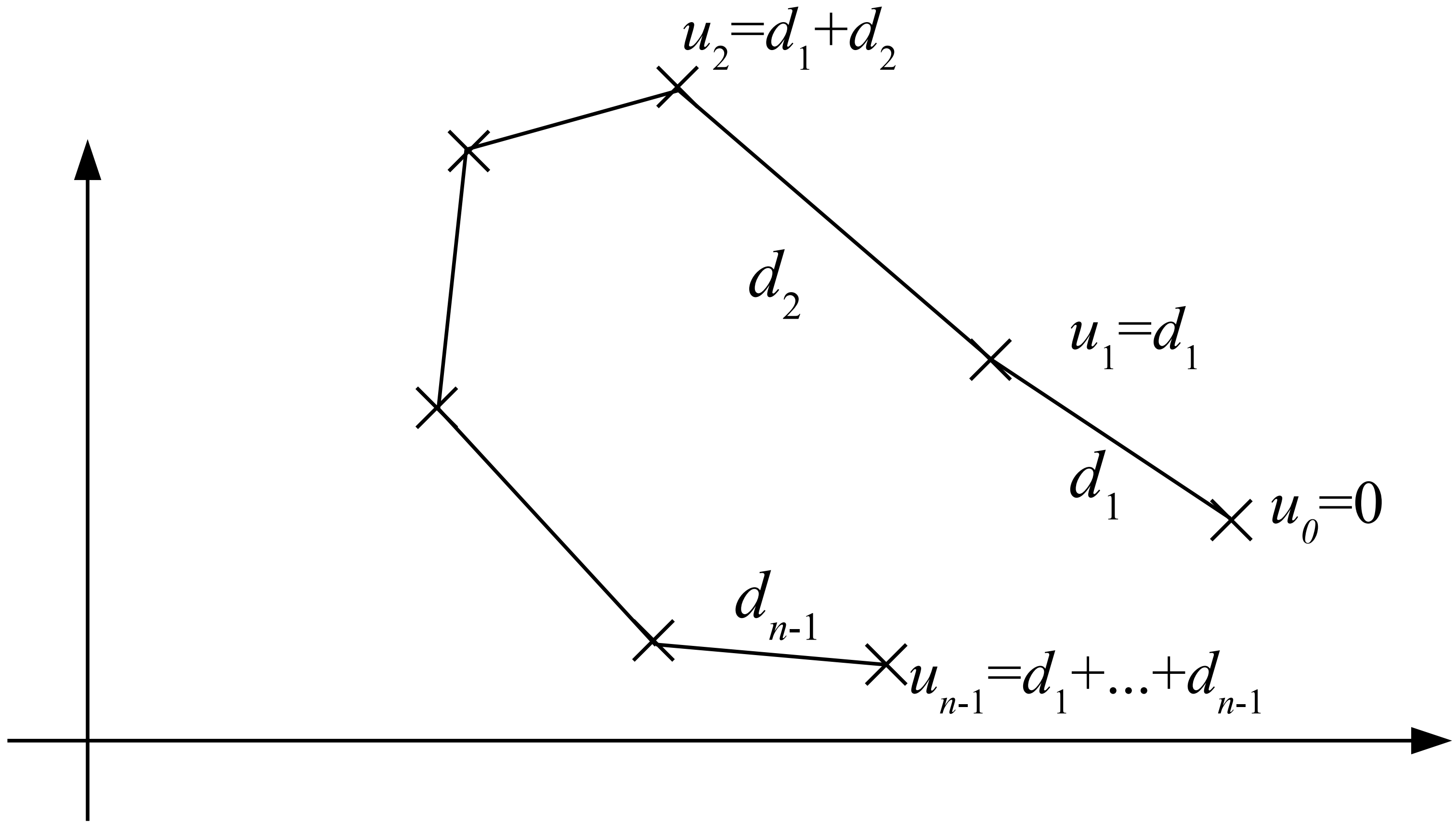
- Or non-uniform



Interpolation

- Can we choose u as a curvilinear abscissa ?
  - In principle no since we don't know the final shape of the curve beforehand (with the exception of interpolation points)
  - We will see later that it is often impossible that u corresponds with s exactly along the whole curve using analytical functions.
  - But nothing forbids to get close to that – numerically...

- Parametrization as an approximate arc length



- In all cases, we want to have :

$$P(u_i)=P_i \equiv \begin{cases} x(u_i)=x_i \\ y(u_i)=y_i \end{cases}$$

- We are going to interpolate the functions $x(u)$ and $y(u)$ with ONE polynomial with n parameters
  - This one must be of order $p=n$-1 :

$$x(u)=a_0+a_1u+a_2u^2+\cdots+a_{n-1}u^{n-1}=\sum_{j=0}^{n-1} a_j u^j$$

  - We set the linear system and solve...

$$\begin{cases} x(u_1)=x_1 \\ \vdots \\ x(u_{n-1})=x_{n-1} \end{cases}$$

- Vandermonde matrix

$$\begin{cases} x(u_1)=x_1 \\ \vdots \\ x(u_{n-1})=x_{n-1} \end{cases} \longrightarrow \begin{pmatrix} 1 & u_0 & u_0^2 & \cdots & u_0^{n-1} \\ 1 & u_1 & u_1^2 & \cdots & u_1^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & u_{n-1} & u_{n-1}^2 & \cdots & u_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

Can be solved by classical numerical methods, but ...

- The condition number of this matrix is **VERY** bad
- It must be solved for each RHS member ((x_i) or (y_i)) , or have to take the inverse of this matrix, or perform an LU decomposition.

Interpolation

- Instead of setting the polynomial in x and in y and solving, we can put it under the following form :

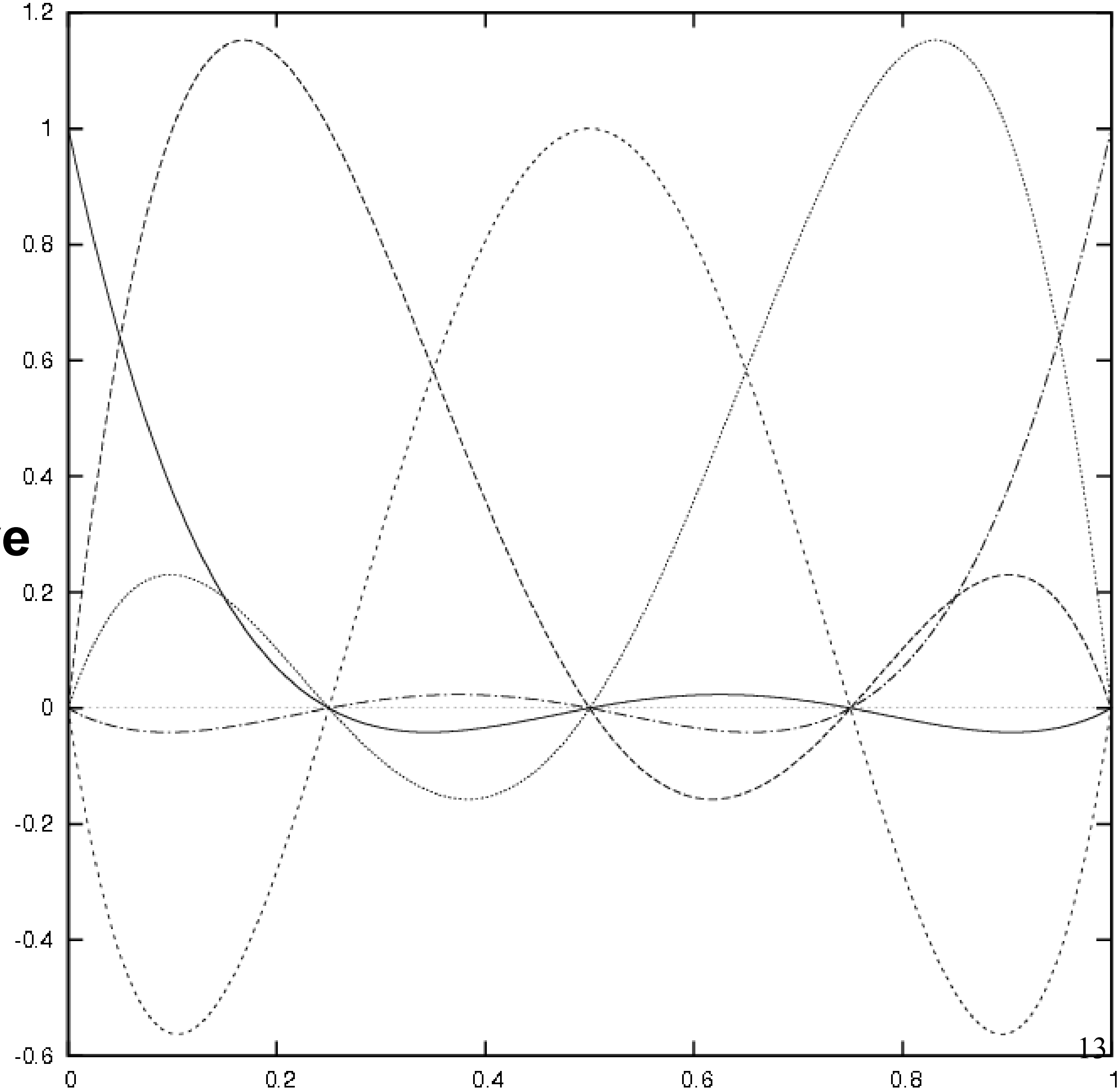
$$\begin{cases} x(u) = \sum_{0}^{n-1} x_i l_i^p(u) \\ y(u) = \sum_{0}^{n-1} y_i l_i^p(u) \end{cases} \Leftrightarrow P(u) = \sum_{0}^{n-1} P_i l_i^p(u)$$

, where the $l_i^p(u)$ are a polynomial basis of order $p=n-1$.

  - These polynomials verify, for an interpolation : $l_i^p(u_j) = \delta_{ij}$
  - We have only one computation to do for any position of the interpolation points, knowing the u_i (the parametrization)

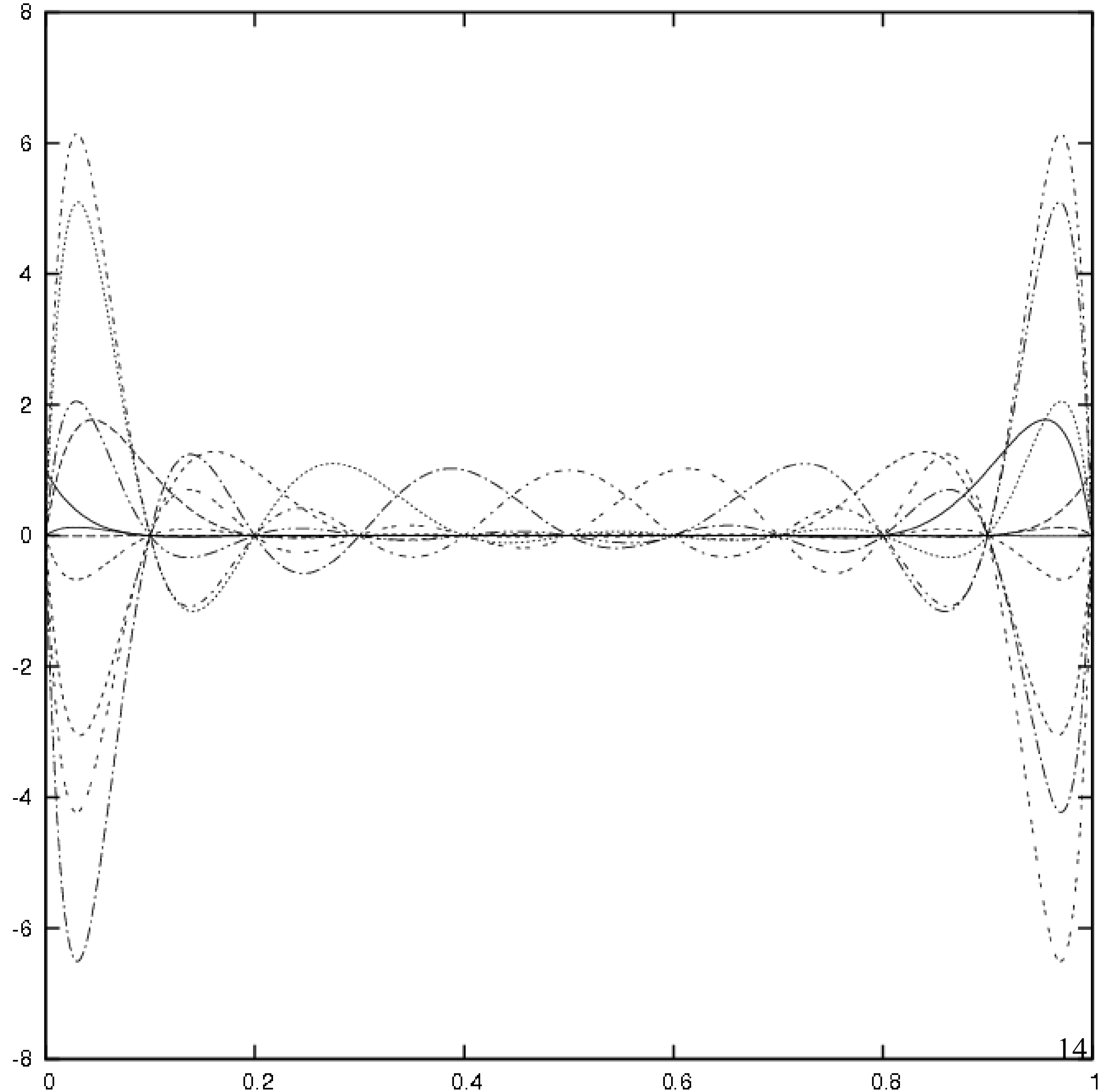
Lagrange polynomials

- Order 4
- The u_i are evenly distributed between $u=0$ and $u=1$
- The sum is equal to 1 (partition of unity)
- **Presence of negative values**



Lagrange polynomials

- Order 10
- Presence of huge overshoots near the boundaries



Lagrange polynomials

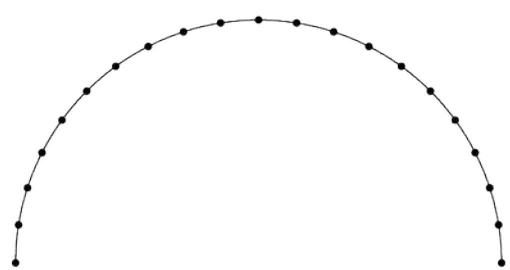
- The interpolation is represented by the following form :

$$P(u)=\sum_{i=0}^{n-1} P_i l_i^p(u) \quad \text{with} \quad l_i^p(u)=\prod_{j=0,i\neq j}^{n-1} \frac{(u-u_i)}{(u_j-u_i)}$$

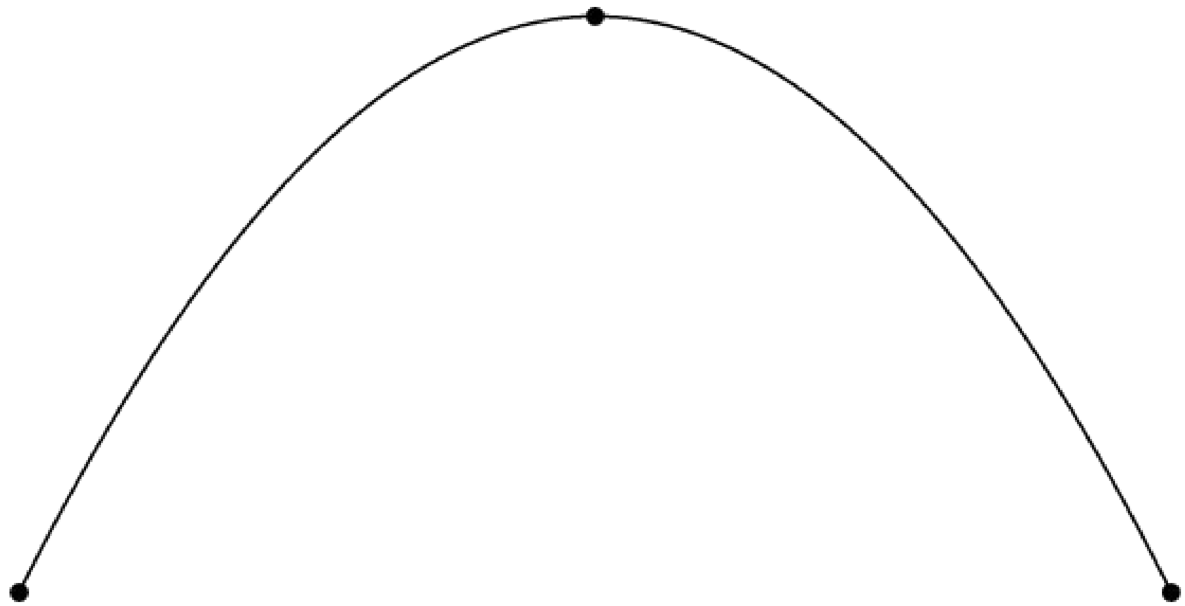
- Two things worth noting:
  - The curve depends **linearly** on the position of the points
  - It is formed by a weighted sum of **basis functions** that express the influence of each point on the curve

Lagrange polynomials

- An experiment
  - We approximate a circle by an increasing number of points
  - Simultaneously , approximation order increases
  - In every case, the curve is C_∞

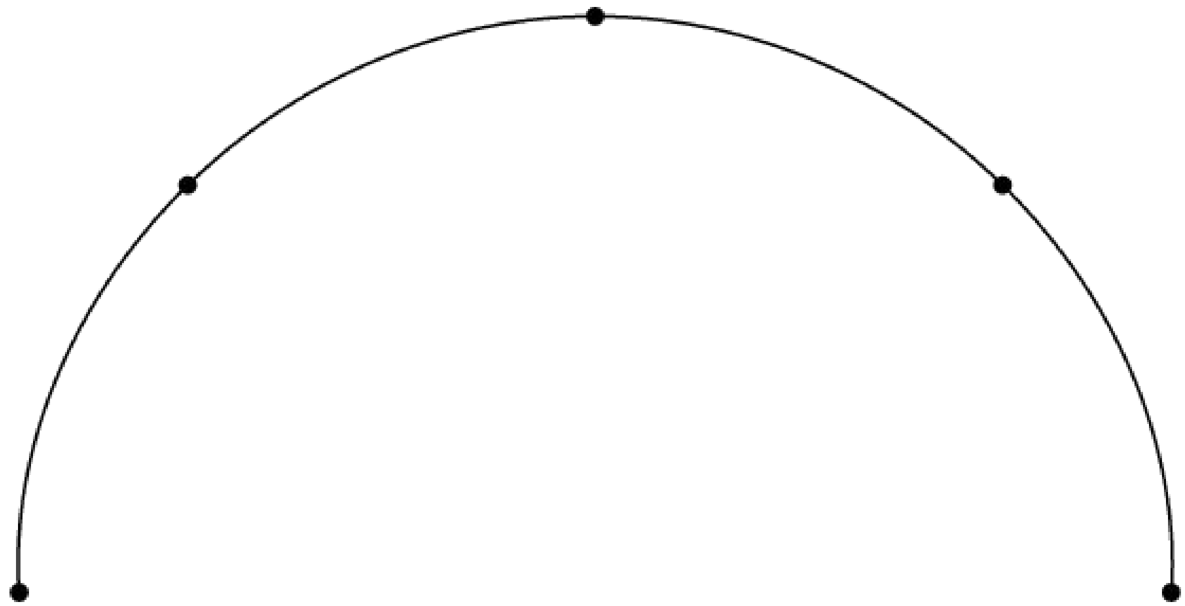


Lagrange polynomials



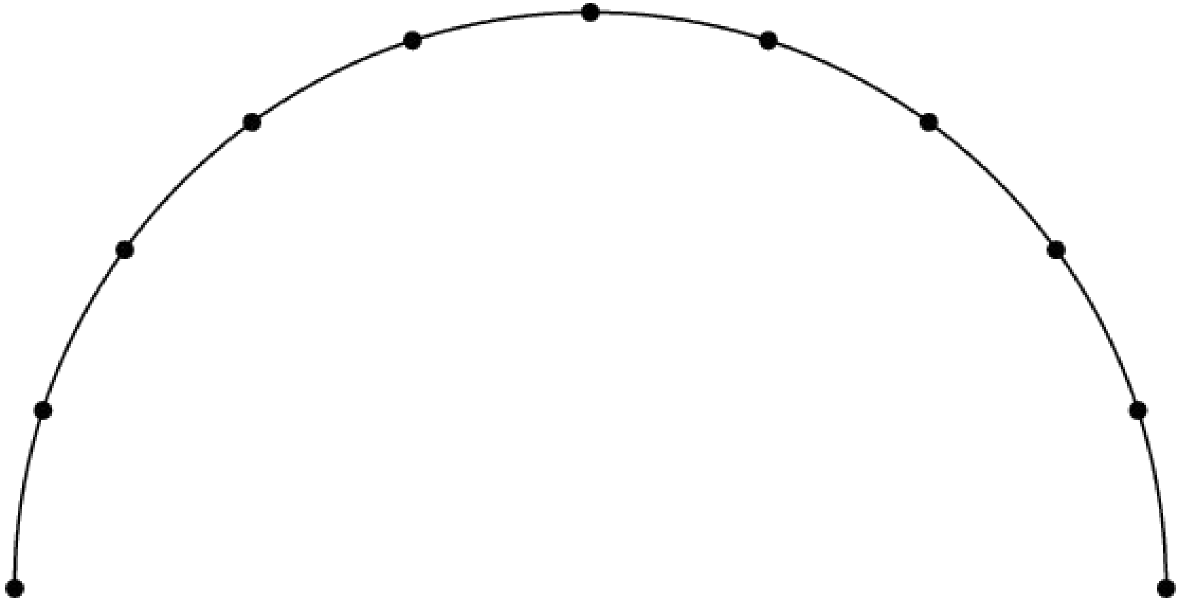
3 points, order 2 (a parabola !)

Lagrange polynomials



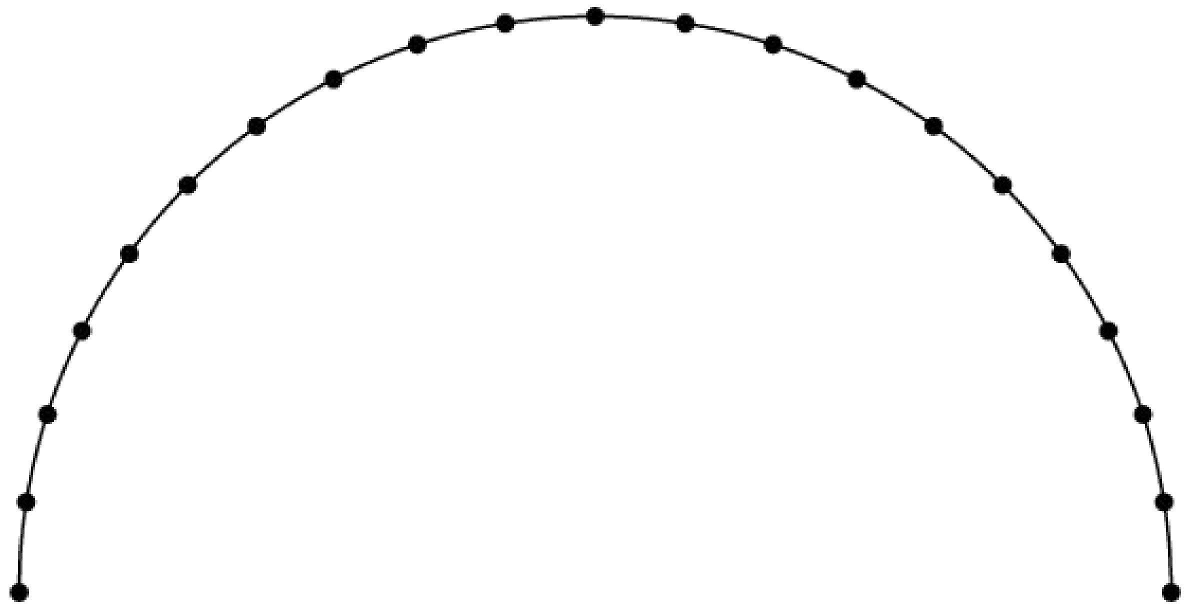
5 points, order 4

Lagrange polynomials



11 points, order 10

Lagrange polynomials



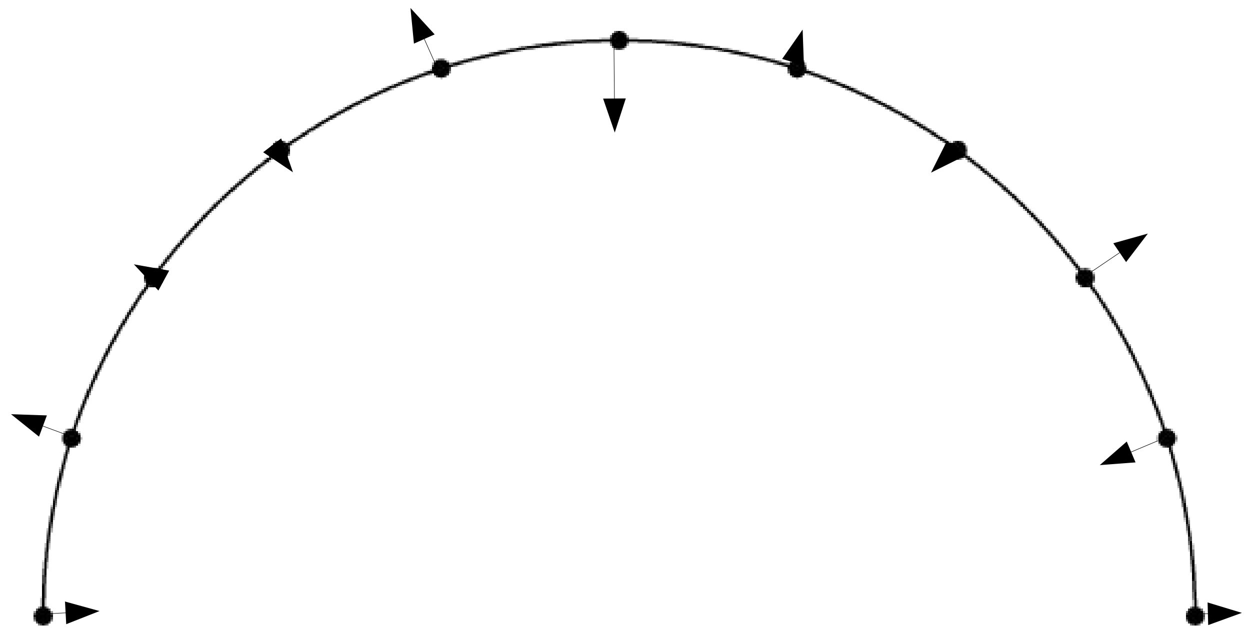
21 points, order 20

Lagrange polynomials

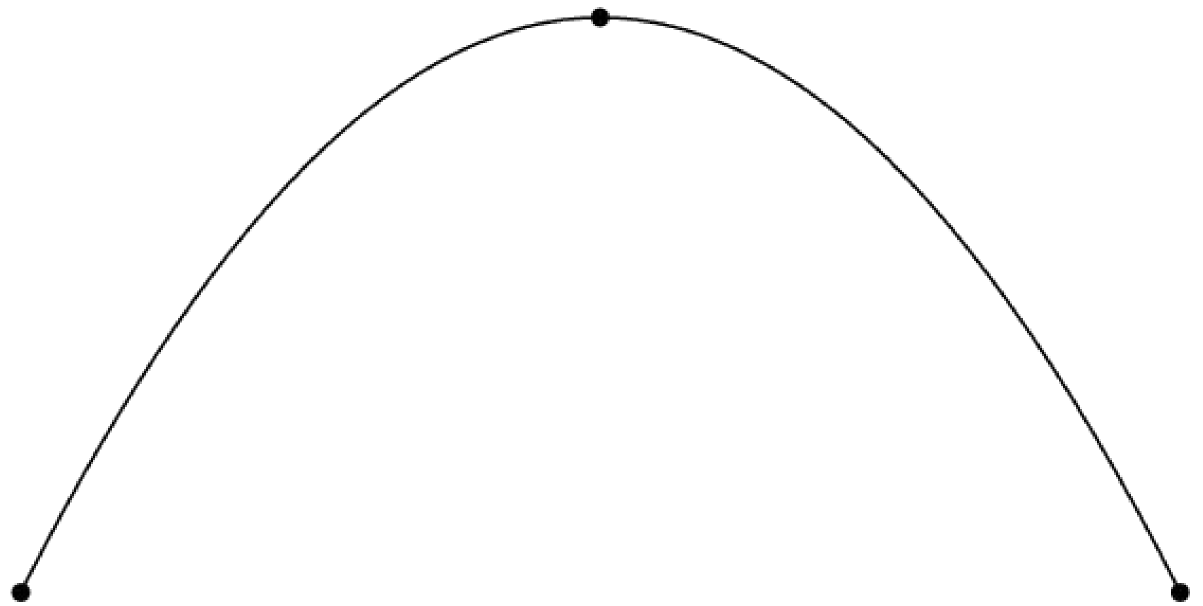
- Does it work ?
- The points were set exactly on the circle
  - What occurs if their position is inaccurate ?
  - Or if the approximated shape is not so simple ?
  - We are going to see two cases
    - The coordinates of the points are perturbed randomly
    - A deterministic increase and decrease of the radius

Lagrange perturbation

- Random perturbation
  - Each point is moved radially by a value between -0.5 and +0.5 % of the circle's radius

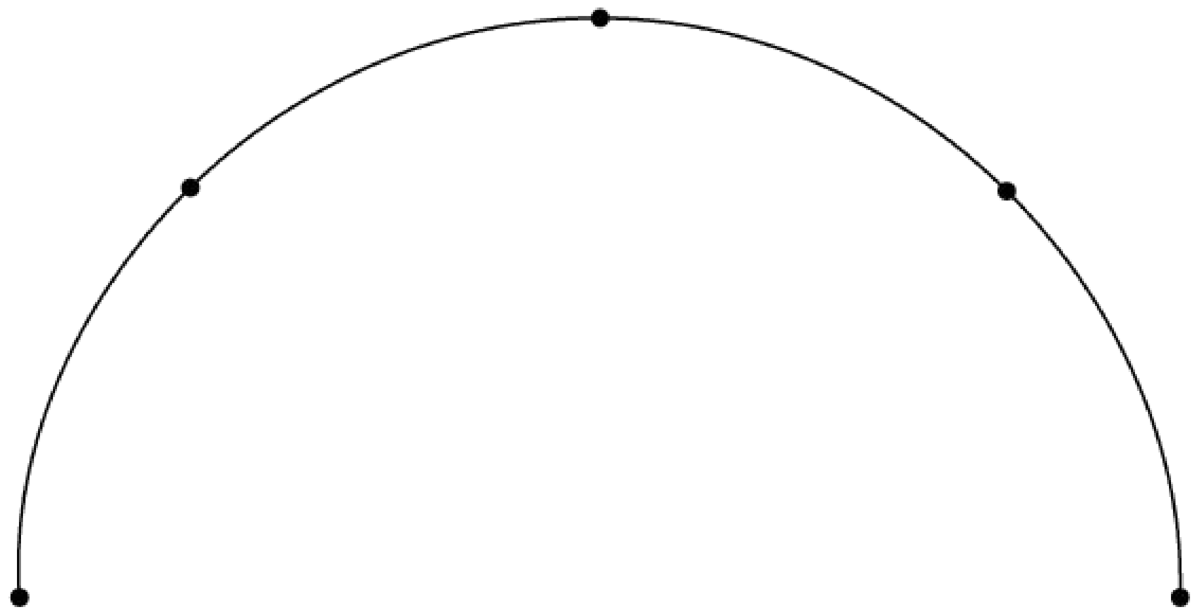


Lagrange polynomials



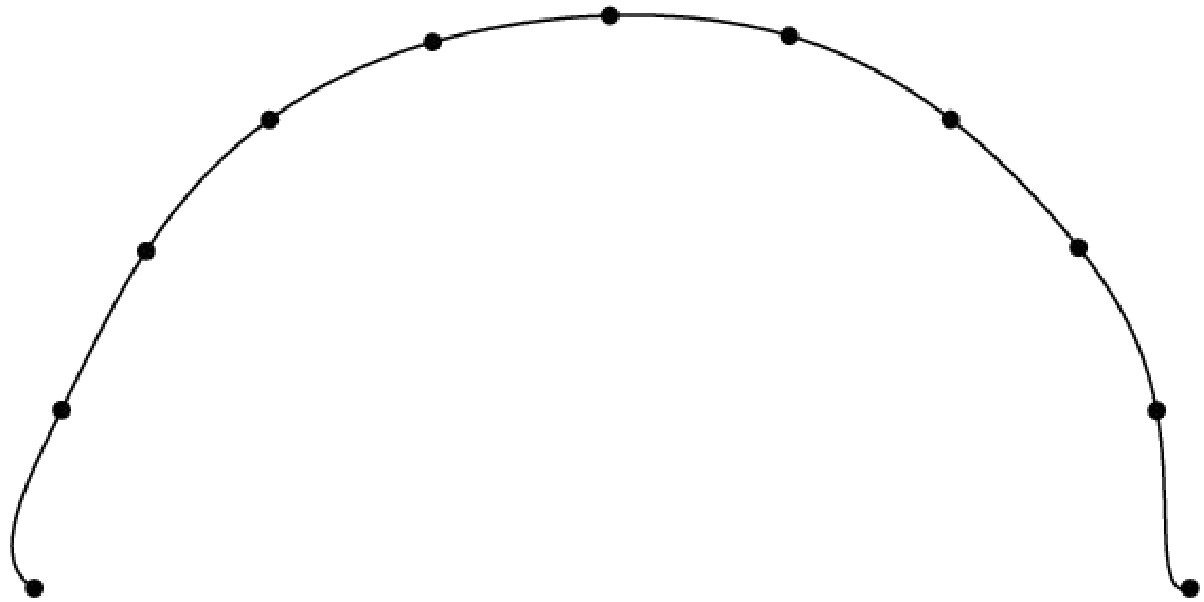
3 points, random perturbation 1%

Lagrange polynomials



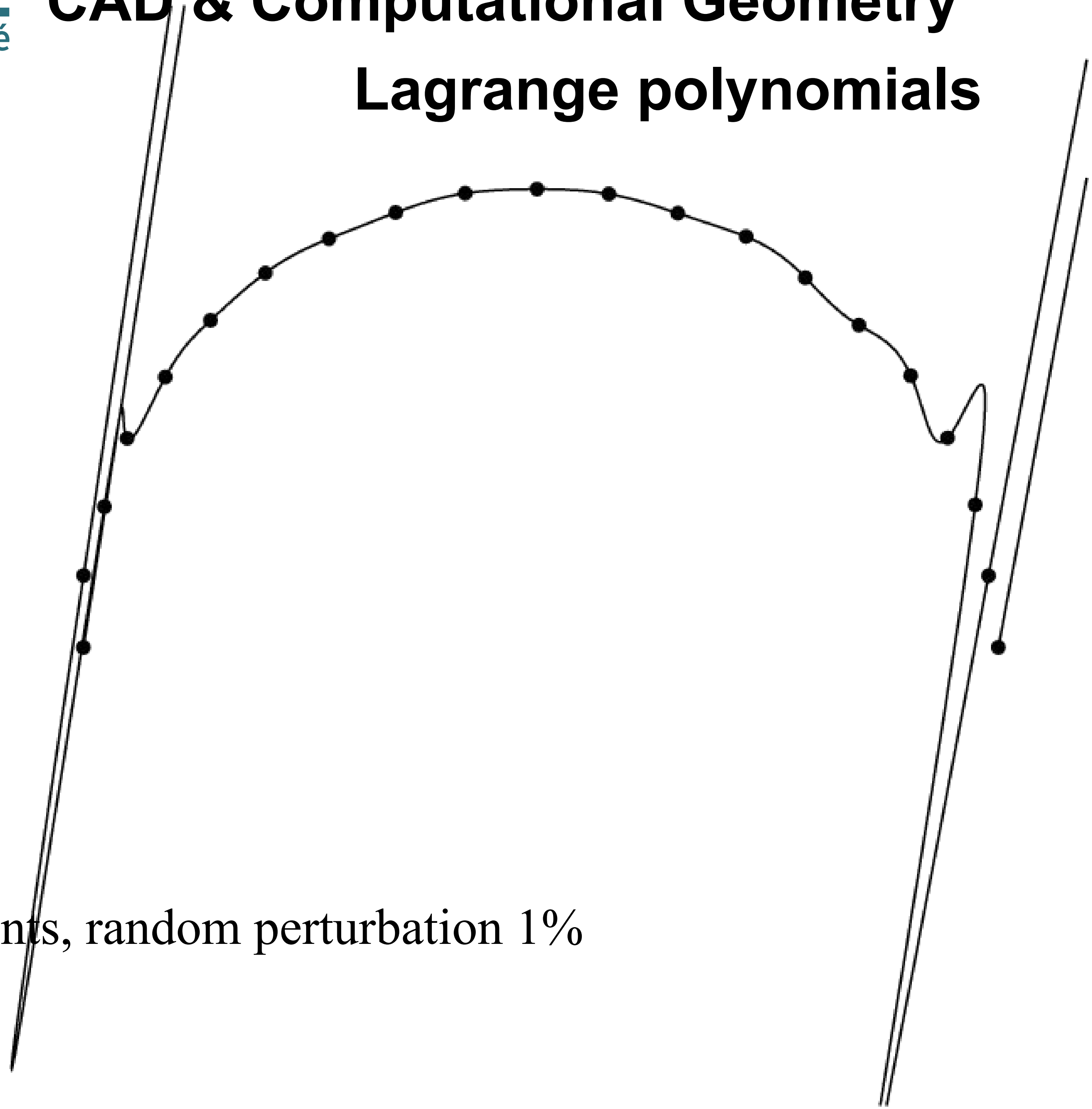
5 points, random perturbation 1%

Lagrange polynomials



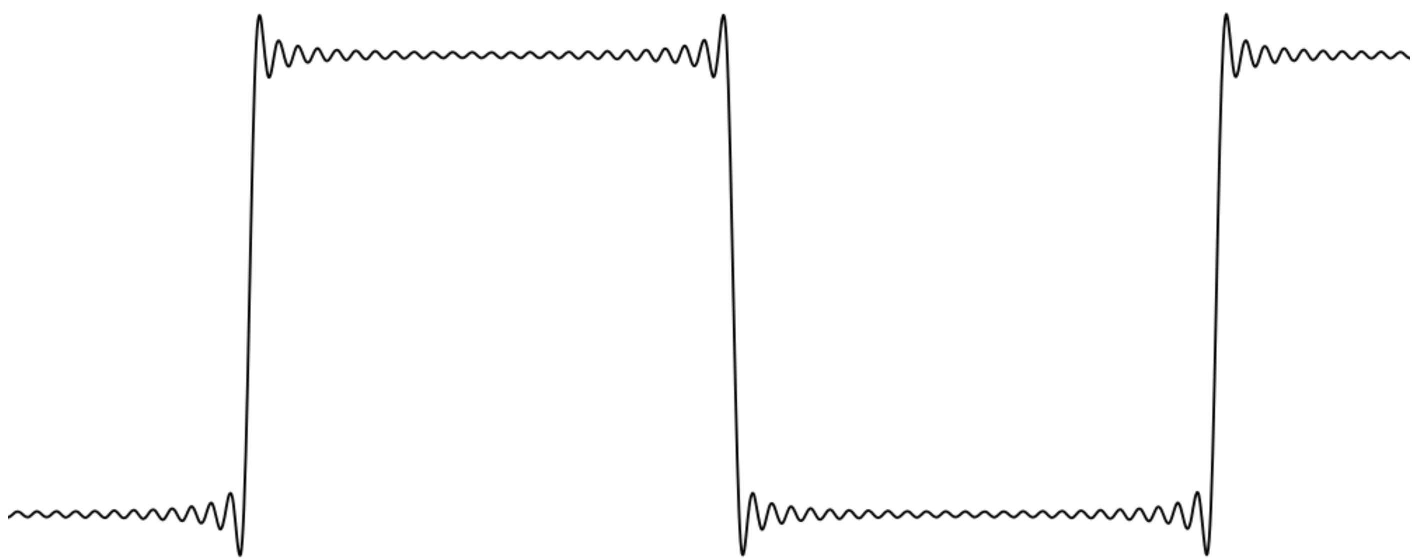
11 points, random perturbation 1%

Lagrange polynomials



21 points, random perturbation 1%

- Runge phenomenon
  - Similar to Gibbs phenomenon of the decompositions in harmonic functions



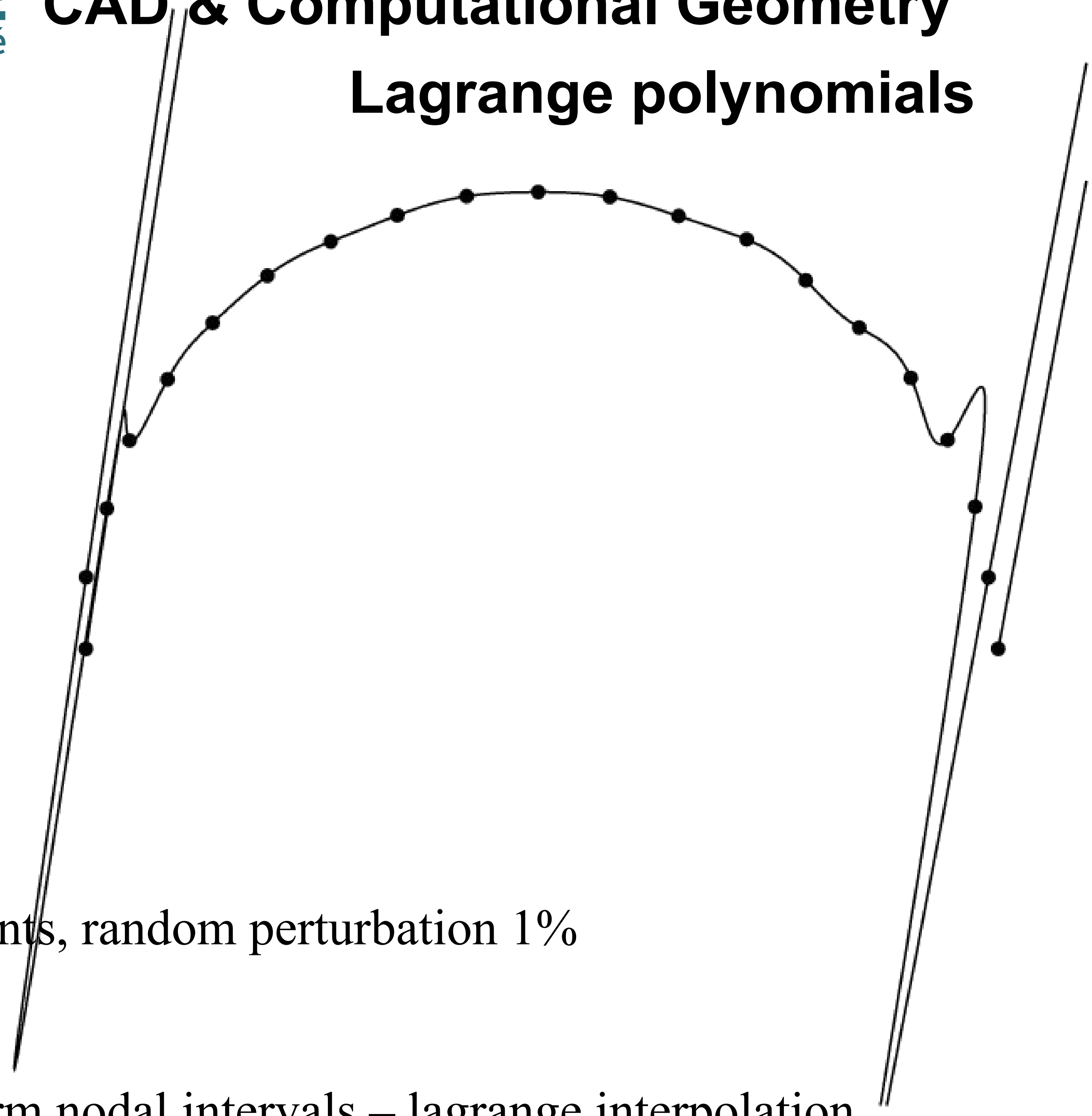
Lagrange polynomials

- How to minimize Runge's phenomenon ?
  - The problem here is the use of a unique polynomial and regular intervals between knots.
  - Instead, if we concentrate knots at the extremities, the interpolation is less prone to Runge's phenomenon.
  - Make use of Chebyshev knots:

$$u_i^n = \cos\left(\frac{2i-1}{2n}\pi\right)$$ in the interval $[-1,1]$

or $$u_i^n = \frac{1}{2} + \frac{1}{2}\cos\left(\frac{2i-1}{2n}\pi\right)$$ in the interval $[0,1]$

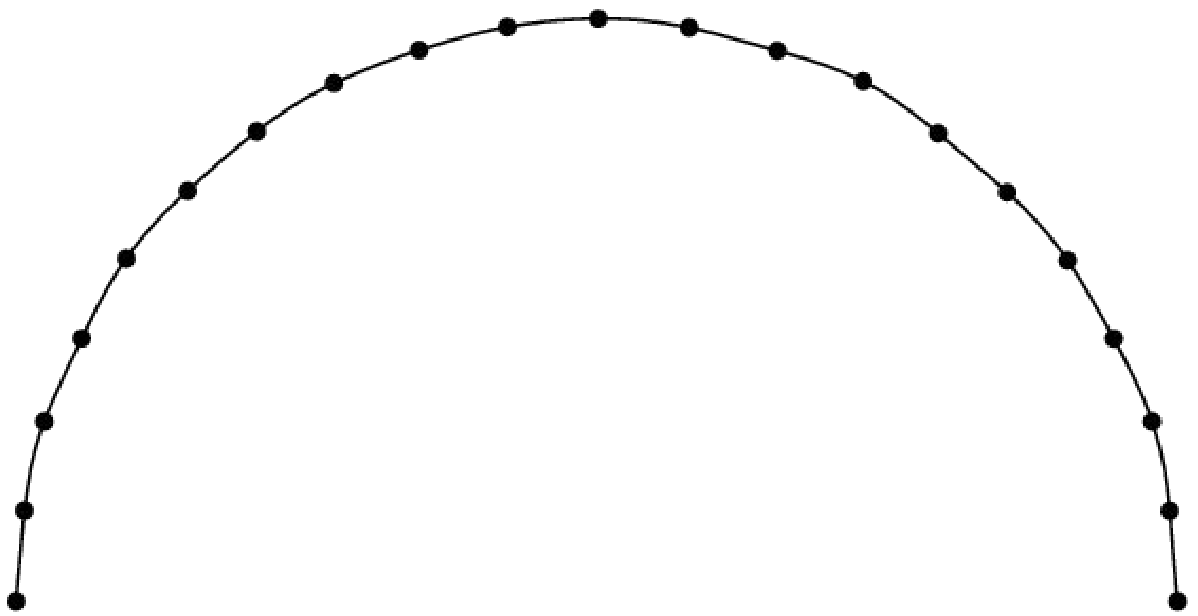
Lagrange polynomials



21 points, random perturbation 1%

Uniform nodal intervals – lagrange interpolation

Lagrange polynomials



21 points, random perturbation 1%

Using Chebychev knots

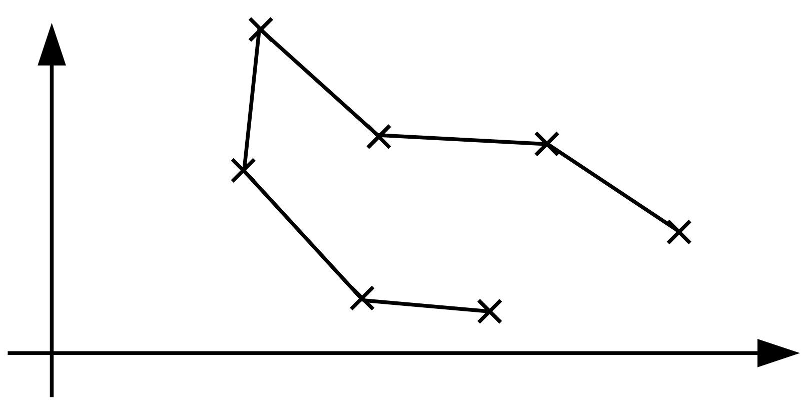
Lagrange polynomials

Morality :

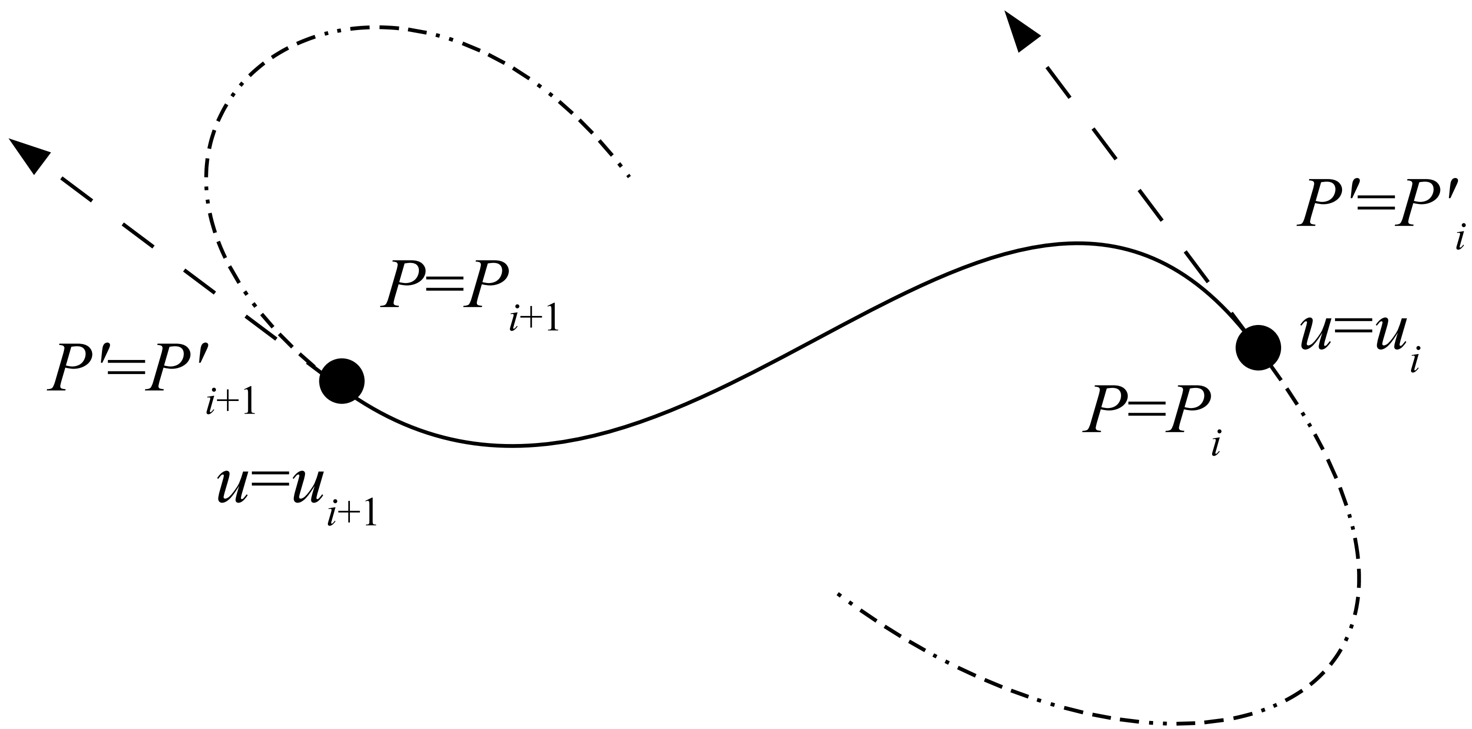
- Lagrange interpolation is not suited beyond 10's of control points because of the Runge phenomenon
- A modification of the position of a control point leads to global changes of the curve.
- (The evaluation of high order polynomials expressed as monomials leads to numerical problems.)
- No control of the slopes at the boundary of the curve (start and finish).

Splines

- Motivation
  Let us imagine that we have many (100's of) control points
  - But we don't want a Lagrange interpolation !
  - We should stay with a low order scheme but conserve enough freedom to pass through every point
    - Curve defined by pieces ... and of low order (1)



- We are going to build a low order interpolation for each knot interval, such that we can impose slopes at the knots.



- In each range $[i,i+1]$, we want to have an independent polynomial
- We have 4 parameters : position at each knot and associated tangents.
  - The basis must have 4 degrees of freedom, thus be of order 3 in the case of polynomials.

$$P(u_i)=P_i \equiv \begin{cases} x(u_i)=x_i \\ y(u_i)=y_i \\ \quad \ldots \end{cases}$$

$$x_{[i]}(u)=A_{[i]0}+A_{[i]1}u+A_{[i]2}u^2+A_{[i]3}u^3 \quad , u\in[u_i , u_{i+1}]$$

Splines

- First, every interval has a unit length i.e.

$$u_{i+1}-u_i=1$$

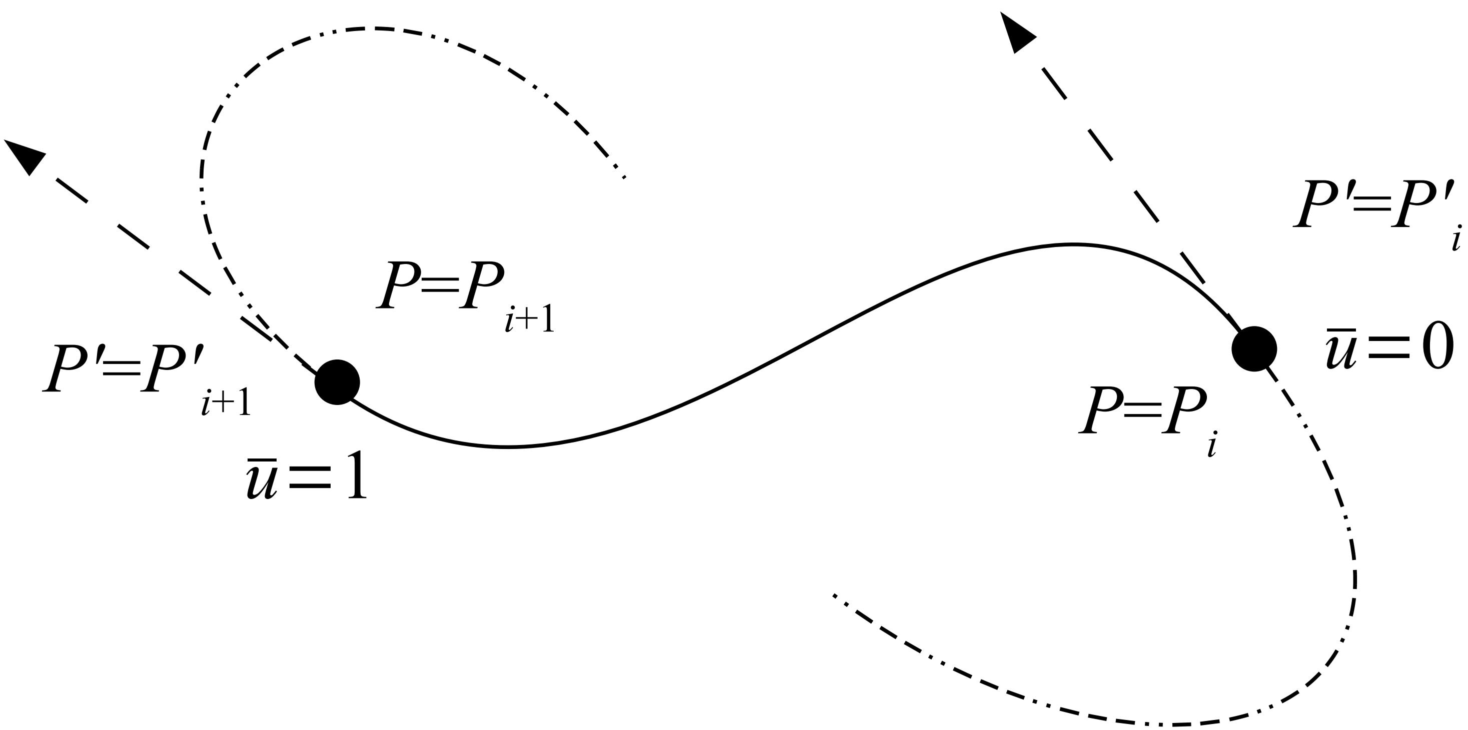
- Then we ensure identical intervals $[0...1]$ between each interpolation point :

$$\bar{u}=\frac{u-u_i}{u_{i+1}-u_i}=u-u_i \qquad \frac{d\,\bar{u}}{du}=1$$

- On each interval i , we thus have the following relation:

$$x_{[i]}(\bar{u})=a_{[i]0}+a_{[i]1}\bar{u}+a_{[i]2}\bar{u}^2+a_{[i]3}\bar{u}^3 \quad , \ \bar{u}\in[0,1]$$

Splines



- We pass through both control points:

$$P(\bar{u}=0)=P_i \Leftrightarrow a_{[i]0}+a_{[i]1}\bar{u}+a_{[i]2}\bar{u}^2+a_{[i]3}\bar{u}^3=x_i$$

$$P(\bar{u}=1)=P_{i+1} \Leftrightarrow a_{[i]0}+a_{[i]1}\bar{u}+a_{[i]2}\bar{u}^2+a_{[i]3}\bar{u}^3=x_{i+1}$$

- We impose both slopes :

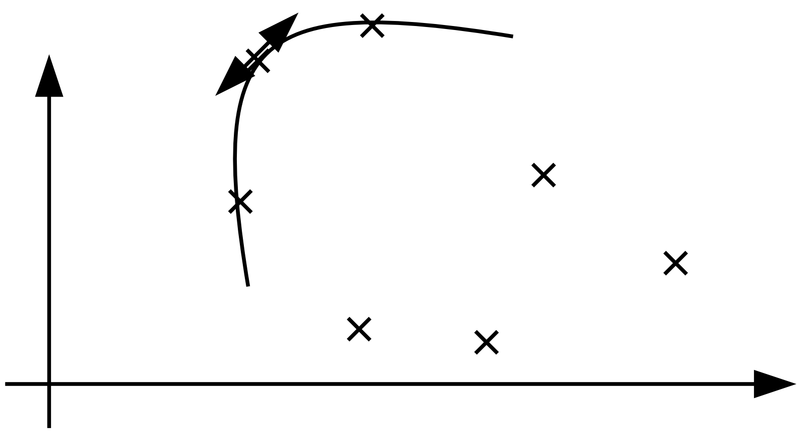
$$P'(\bar{u}_0=0)=P'_i \Leftrightarrow a_{[i]1}+2a_{[i]2}\bar{u}+3a_{[i]3}\bar{u}^2=x'_i$$

$$P'(\bar{u}=1)=P'_{i+1} \Leftrightarrow a_{[i]1}+2a_{[i]2}\bar{u}+3a_{[i]3}\bar{u}^2=x'_{i+1}$$

- At the end :

$$\begin{cases} a_{[i]0}=x_i \\ a_{[i]1}=x'_i \\ a_{[i]2}=3(x_{i+1}-x_i)-2x'_i-x'_{i+1} \\ a_{[i]3}=2(x_i-x_{i+1})+x'_i+x'_{i+1} \end{cases}$$

- We have continuity
- We have continuity of the derivatives
- But how to choose the slopes ?
  - Let the user choose (“artistic” freedom)
  - Automatically ...



- By finite differences with three points :

$$x_i' = \frac{x_{i+1} - x_i}{2(u_{i+1} - u_i)} + \frac{x_i - x_{i-1}}{2(u_i - u_{i-1})}$$

  - At the boundaries, we use finite differences (asymmetric)

$$x_0' = \frac{x_1 - x_0}{u_1 - u_0} \qquad x_{n-1}' = \frac{x_{n-1} - x_{n-2}}{u_{n-1} - u_{n-2}}$$

  - The result depends on the parametrization !

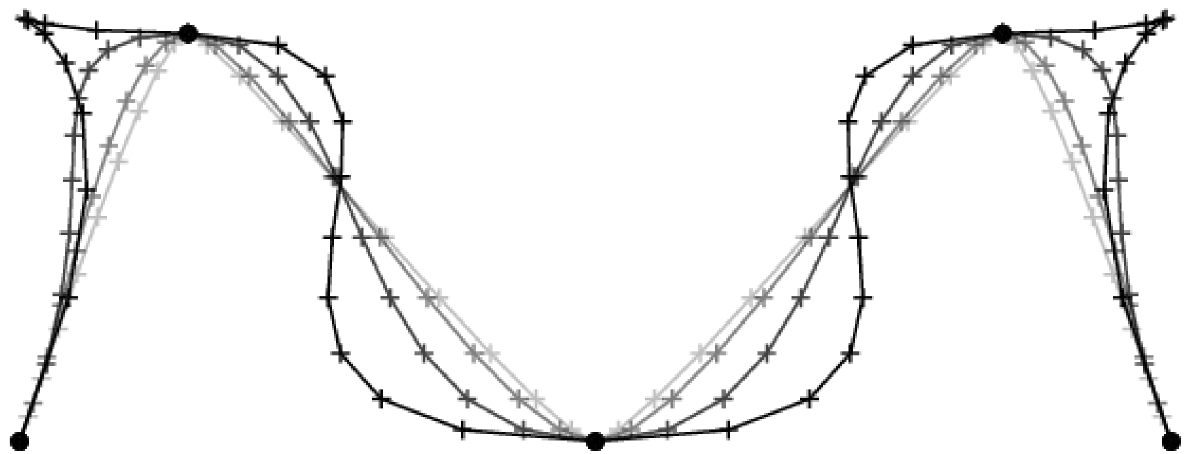
- Cardinal spline

$$x_i' = (1-c)\frac{x_{i+1} - x_{i-1}}{2} \,, \; 0 \le c \le 1$$

$$x_0' = (1-c)(x_1 - x_0)$$
$$x_{n-1}' = (1-c)(x_{n-1} - x_{n-2})$$

  - c is a « tension » parameter. $c=0$ gives yields the so called “Catmull-Rom” spline, $c=1$ a zigzagging line.

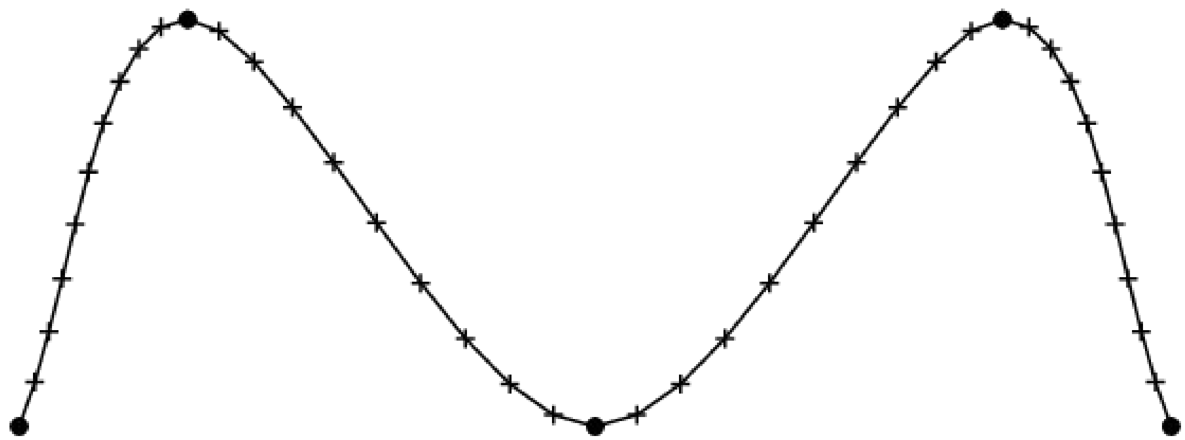
Splines



Continuity of the curve/parameter
but loss of regularity (and of geometric continuity
in many cases)

5 points, finite differences by varying the parametrization
[0..1] , [0..2] , [0..5] , [0..10]

Splines

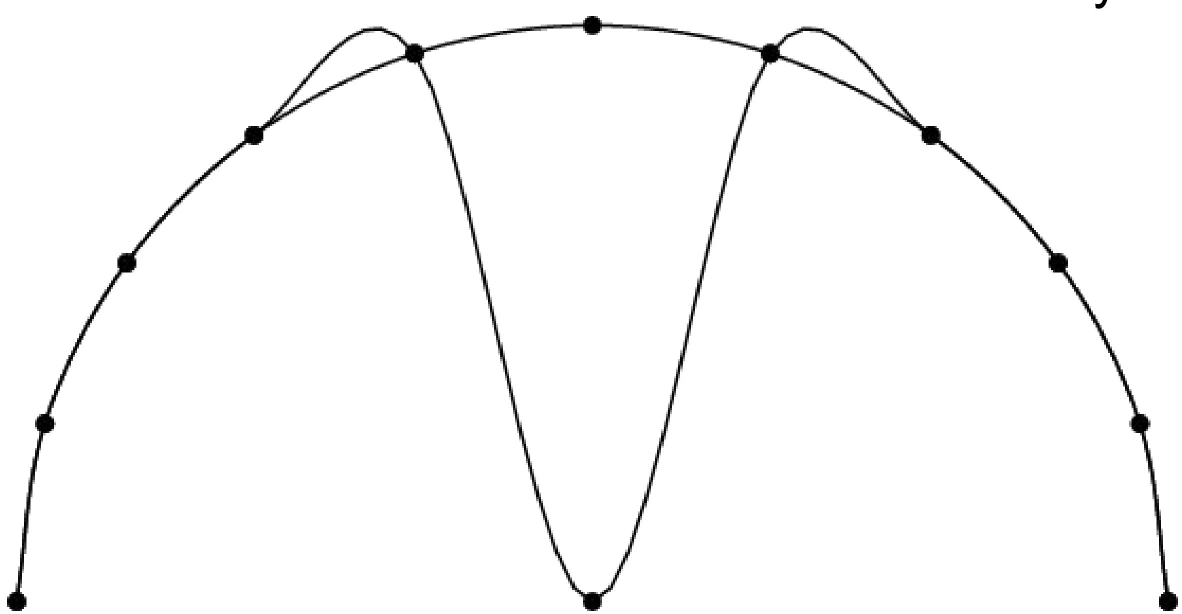


5 points, Cardinal Spline (Catmull-Rom) c=0

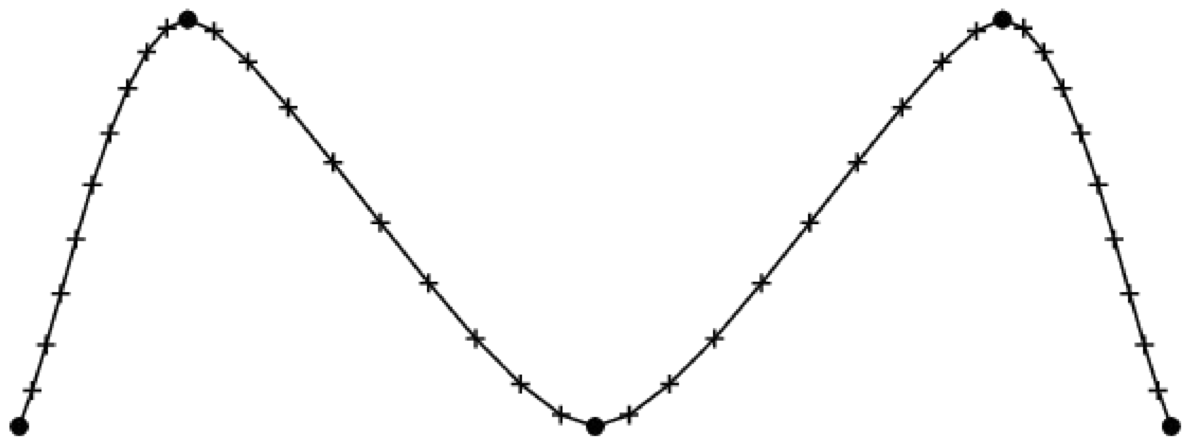
Splines

Catmull-Rom Splines are widely used in computer graphics

- Simple to compute, effective
- Local control (price to pay : discontinuous sec$^{\underline{d}}$ derivative)
- Animations with keyframing
  - Ensures a fluid motion because of the continuity of the slope

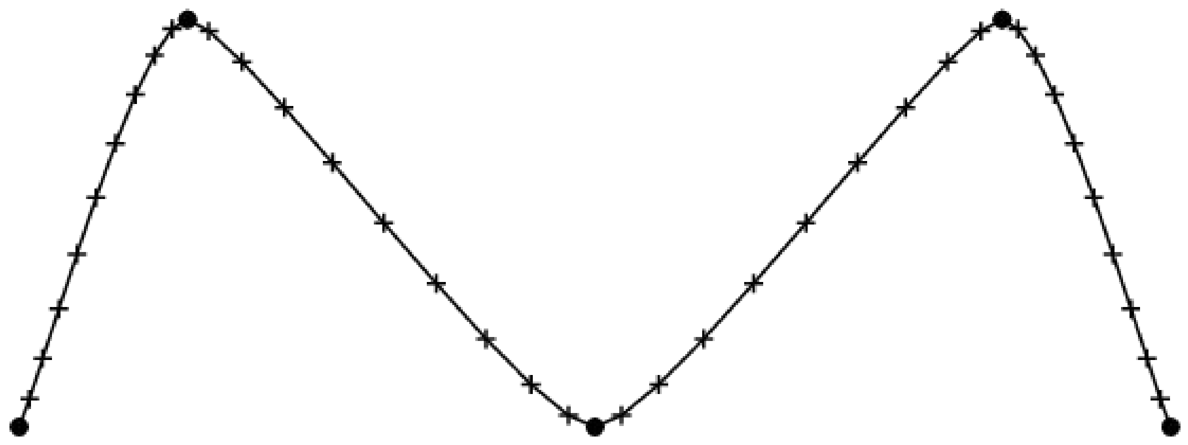


Splines



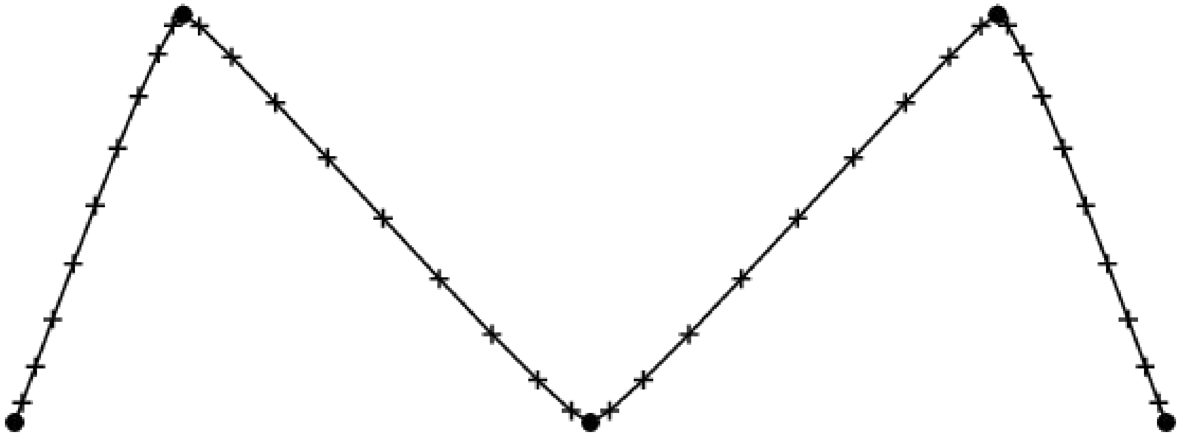
5 points, Cardinal Spline c=0.25

Splines



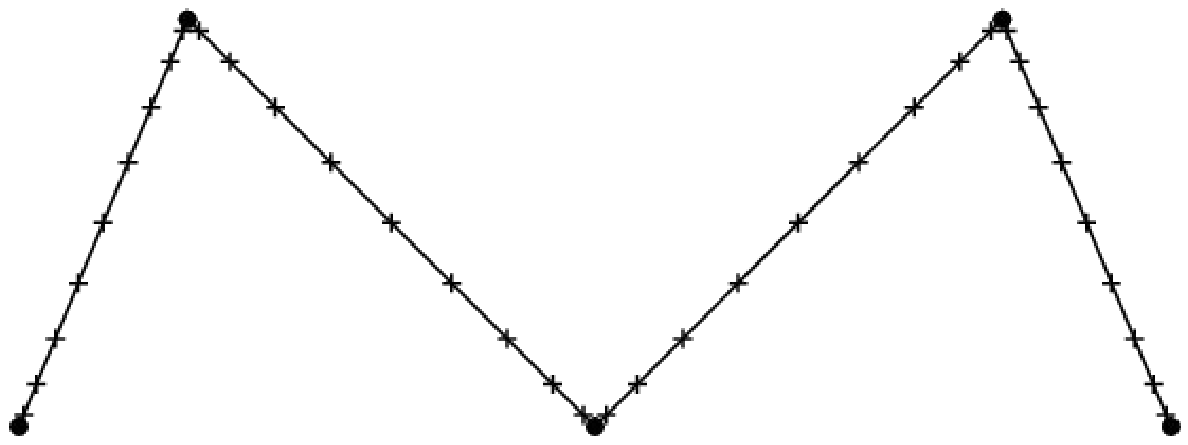
5 points, Cardinal Spline c=0.5

Splines



5 points, Cardinal Spline c=0.75

Splines



5 points, Cardinal Spline c=1.0

Splines

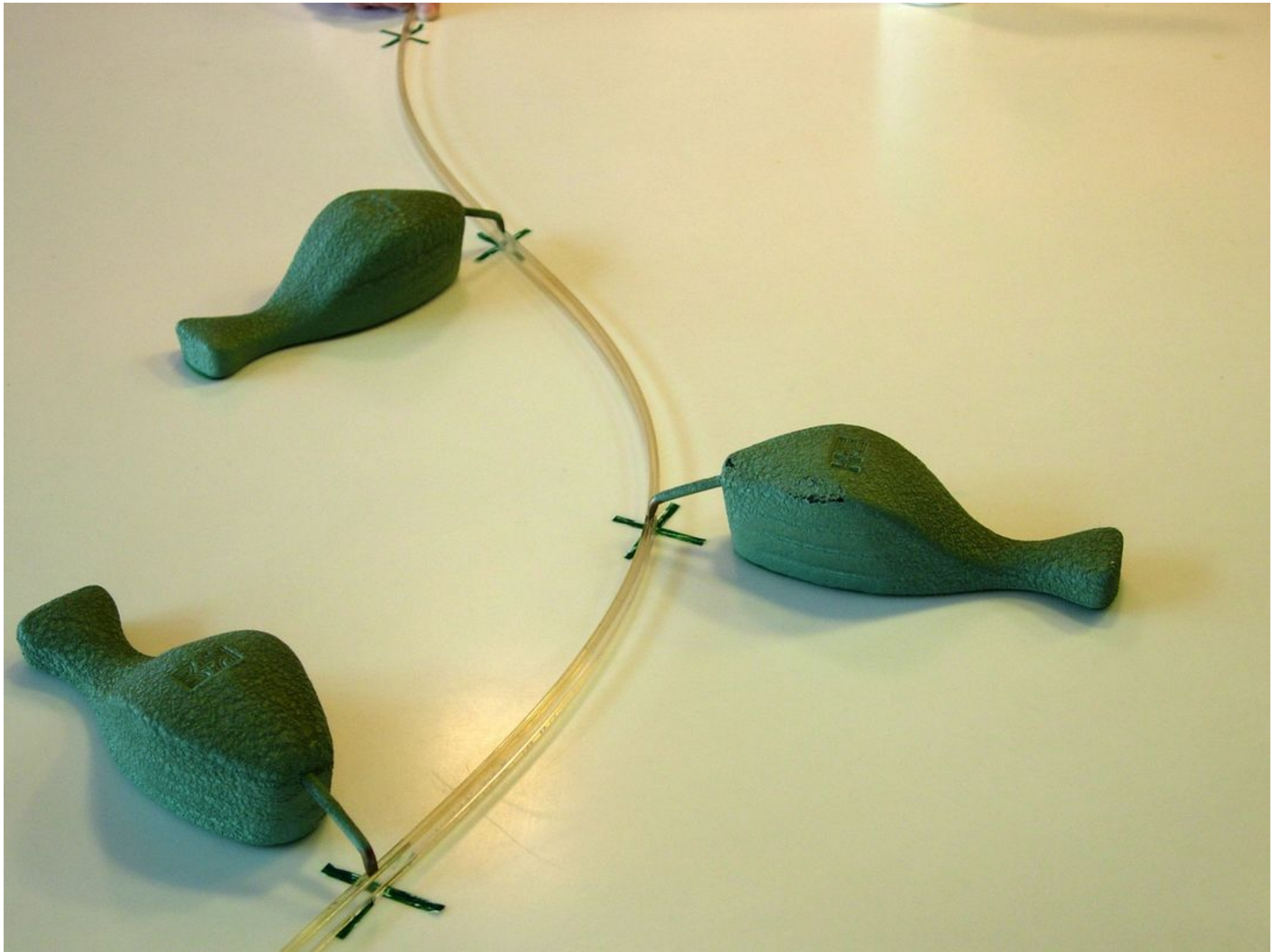
- We can impose the continuity of second derivatives...
  - On a curve with n points, we have n extra relations to impose
  - We may impose the continuity of the second derivative only on the n-2 interior knots

    What about the 2 points on the boundary ?
    - Impose a vanishing second derivative.
      We obtain what is called « **natural spline** »
    - We could also impose the slopes (i.e. only n-2 relations remaining)
    - Or , impose that the third derivative is zero on the points 1 and n-2
      - That means a single polynomial expression for the first two knot intervals, and the last two.

CAD & Computational Geometry

Splines

- Natural Spline : mathematical approximation of the spline historically used in naval construction.



C. DeBoor

Splines

Boeing

- We impose the continuity of the second derivatives

$$x''_{[i-1]}(1)=x''_{[i]}(0) \Leftrightarrow 2\mathrm{a}_{[i-1]2}+6\mathrm{a}_{[i-1]3}=2\mathrm{a}_{[i]2}$$

- We substitute in the “internal” equations

$$2[3(x_i-x_{i-1})-2x'_{i-1}-x'_i]+6[2(x_{i-1}-x_i)+x'_{i-1}+x'_i]$$
$$=2[3(x_{i+1}-x_i)-2x'_i-x'_{i+1}]$$

- Finally we obtain :

$$x'_{i-1}+4x'_i+x'_{i+1}=3(x_{i+1}-x_{i-1})$$

Splines

- At the boundaries we want

$$x''_{[0]}(0)=0 \Leftrightarrow 2a_{[0]2}=0 \qquad 2x'_0+x'_1=3(x_1-x_0)$$

$$x''_{[n-2]}(1)=0 \Leftrightarrow 2a_{[n-2]2}+6a_{[n-2]3}=0 \qquad x'_{n-2}+2x'_{n-1}=3(x_{n-1}-x_{n-2})$$

- We have then a linear system with n unknowns :

$$\begin{pmatrix} 2 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & & \ddots & & & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 2 \end{pmatrix} \begin{pmatrix} x'_0 \\ x'_1 \\ x'_2 \\ \vdots \\ x'_{n-2} \\ x'_{n-1} \end{pmatrix} = \begin{pmatrix} 3(x_1-x_0) \\ 3(x_2-x_0) \\ 3(x_3-x_1) \\ \vdots \\ 3(x_{n-1}-x_{n-3}) \\ 3(x_{n-1}-x_{n-2}) \end{pmatrix}$$

- By solving the system, we have :

$$\begin{pmatrix} x'_0 \\ x'_1 \\ x'_2 \\ \vdots \\ x'_{n-2} \\ x'_{n-1} \end{pmatrix}$$

, which is substituted in

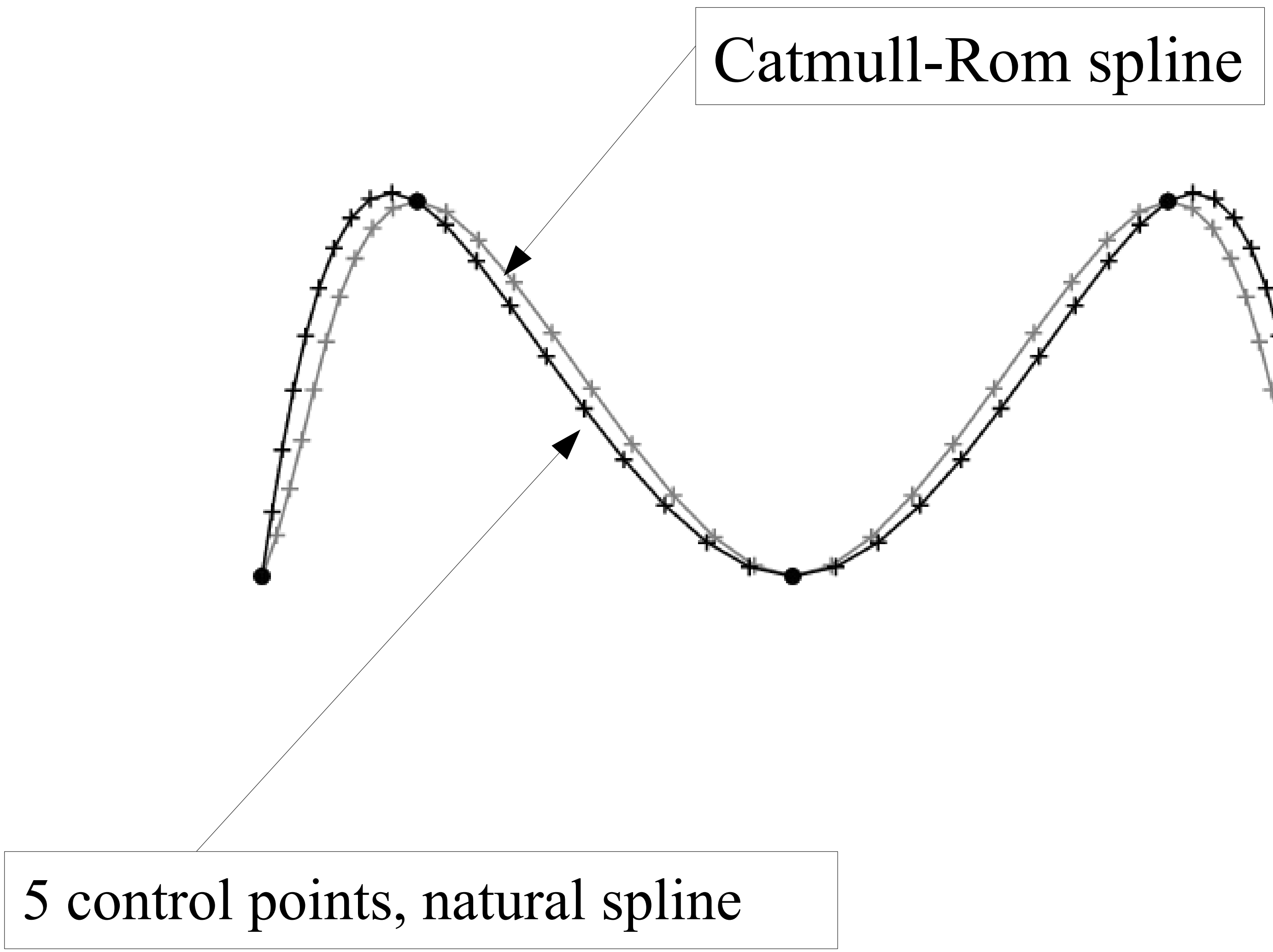
$$\begin{cases} a_{[i]0} = x_i \\ a_{[i]1} = x'_i \\ a_{[i]2} = 3(x_{i+1} - x_i) - 2x'_i - x'_{i+1} \\ a_{[i]3} = 2(x_i - x_{i+1}) + x'_i + x'_{i+1} \end{cases}$$

,to get the polynomial in each portion :

$$x_{[i]}(\bar{u}) = a_{[i]0} + a_{[i]1}\bar{u} + a_{[i]2}\bar{u}^2 + a_{[i]3}\bar{u}^3 \;,\; 0 \le \bar{u} < 1$$

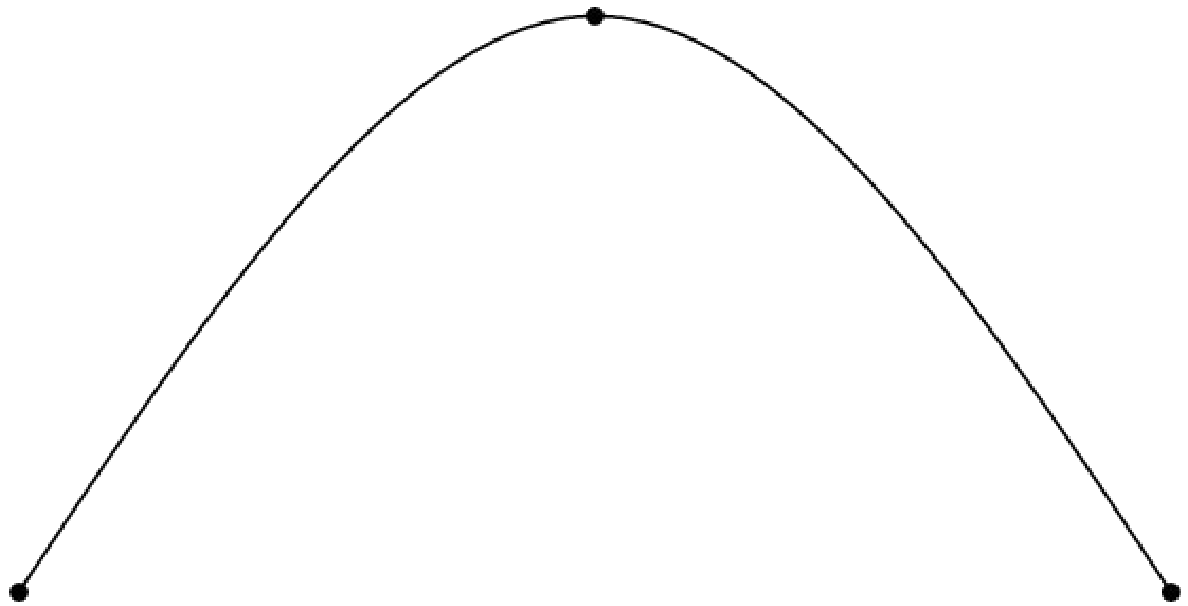
From the global parameter u, we have to find in which portion we are (the value of i) , then compute right polynomial...

Splines



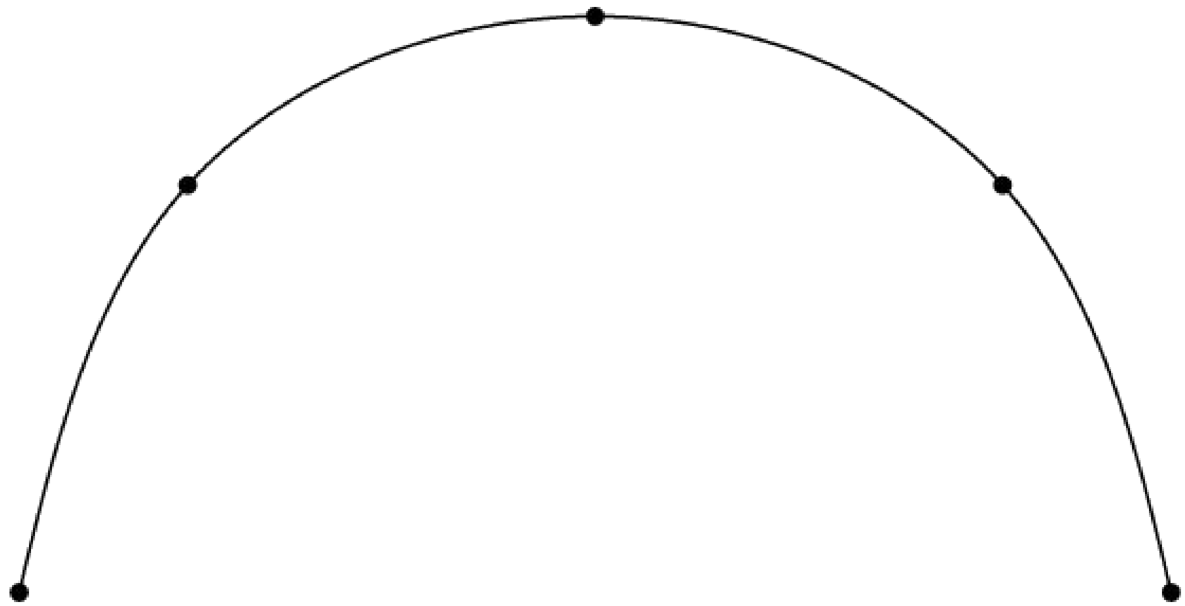
- Another experiment
  - We approximate a circle by a number of increasing points
  - Simultaneously , ~~the order of the approximation~~ the number of pieces increases.
  - In all the cases, the curve is $\cancel{C_\infty}$ C_2

Splines



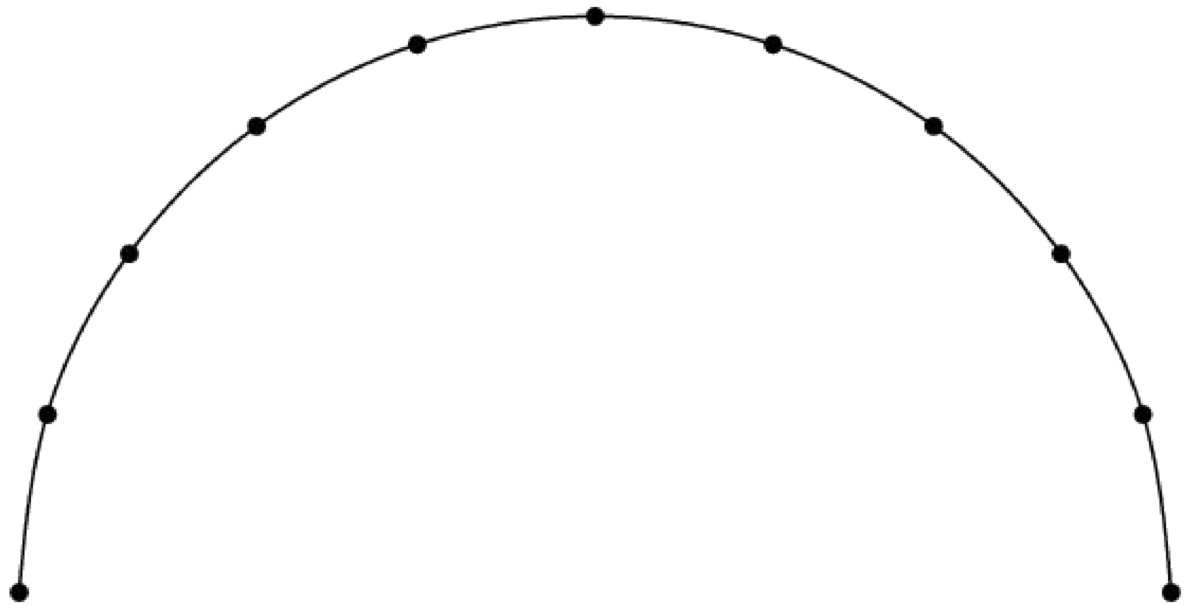
3 points, order 3!

Splines



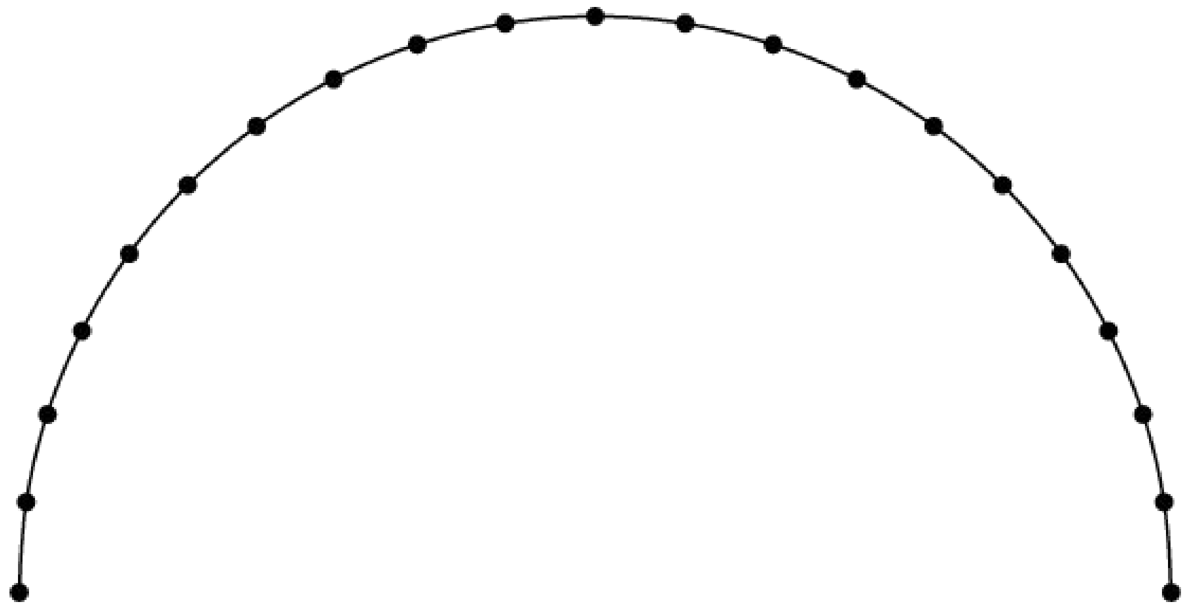
5 points

Splines



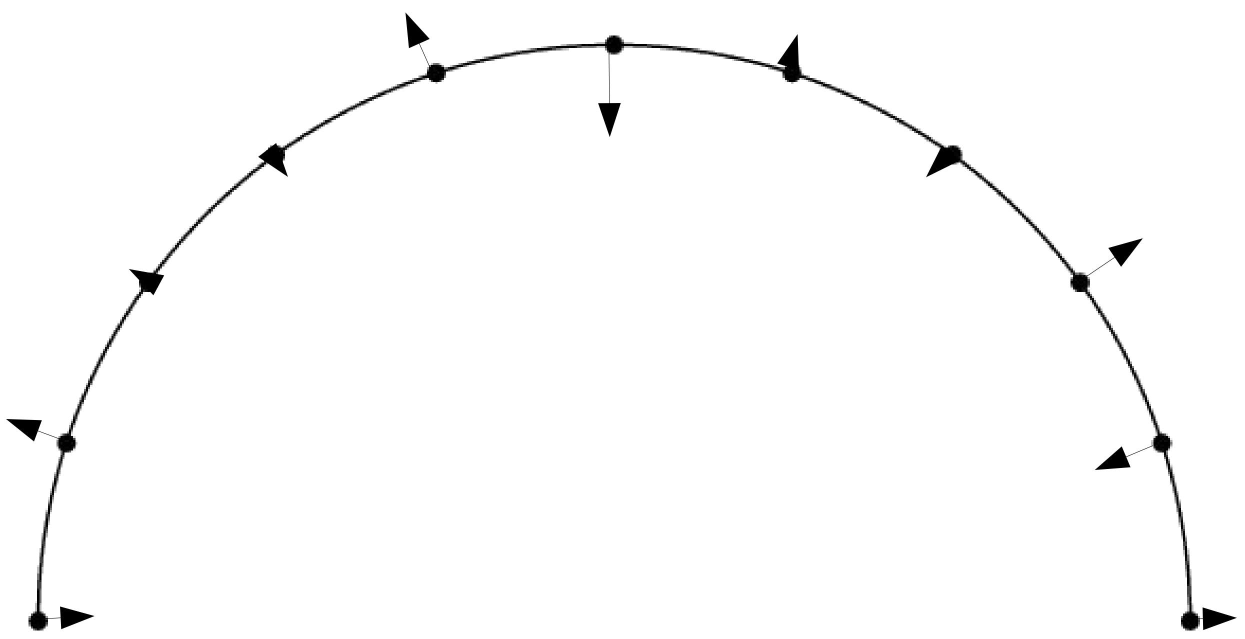
11 points

Splines

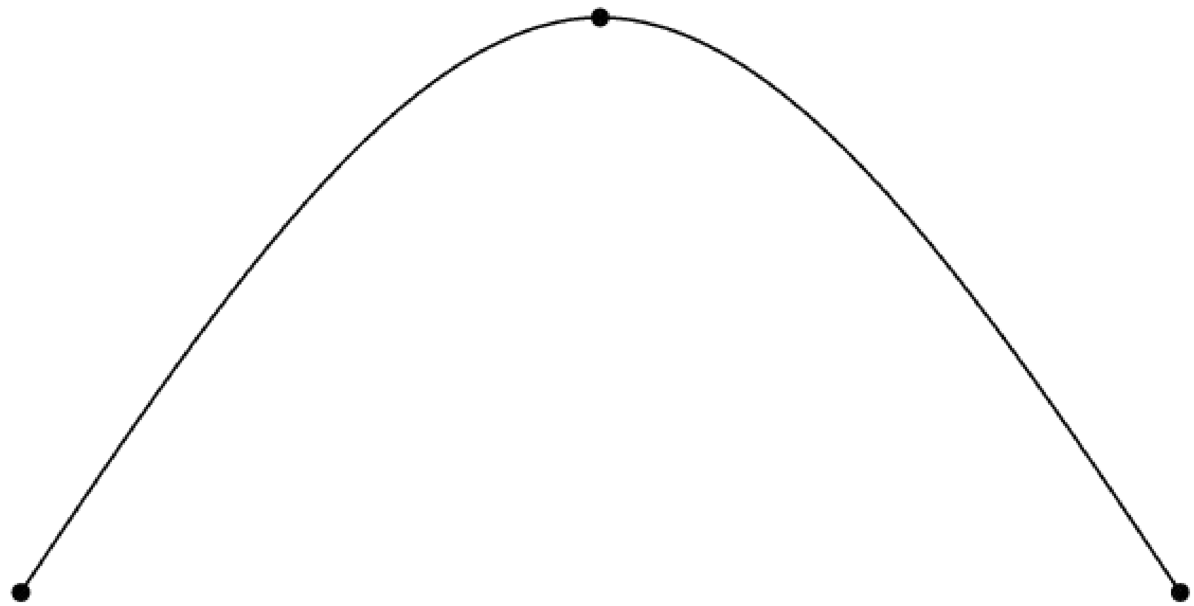


21 points

- Random perturbation
  - Each point is moved radially by a value between -0.5 and +0.5 % of the circle's radius

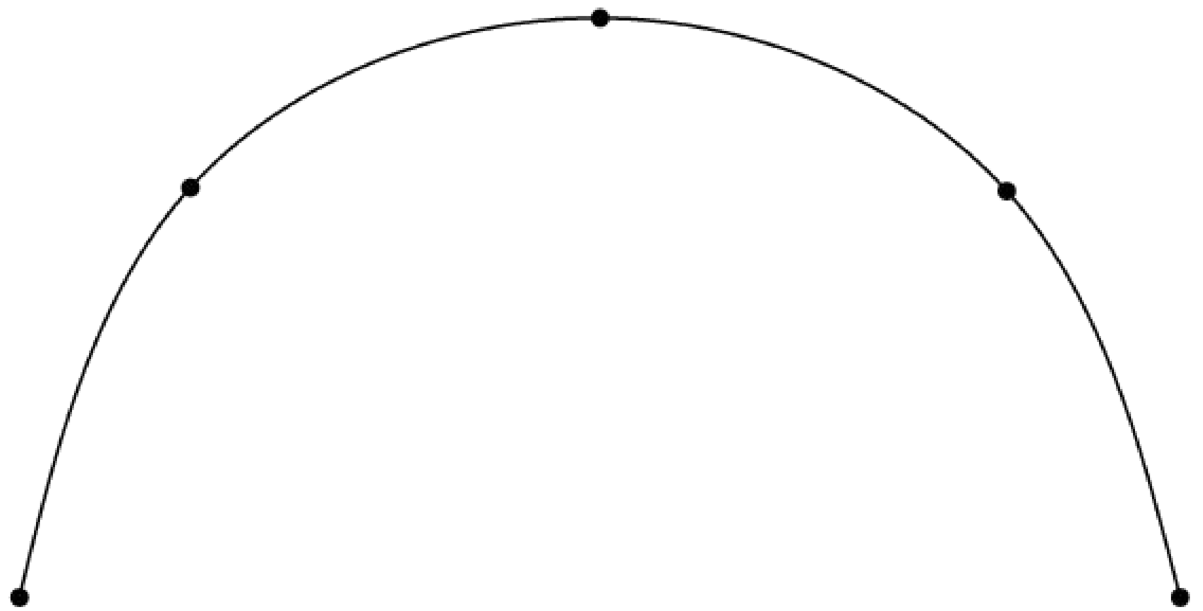


Splines



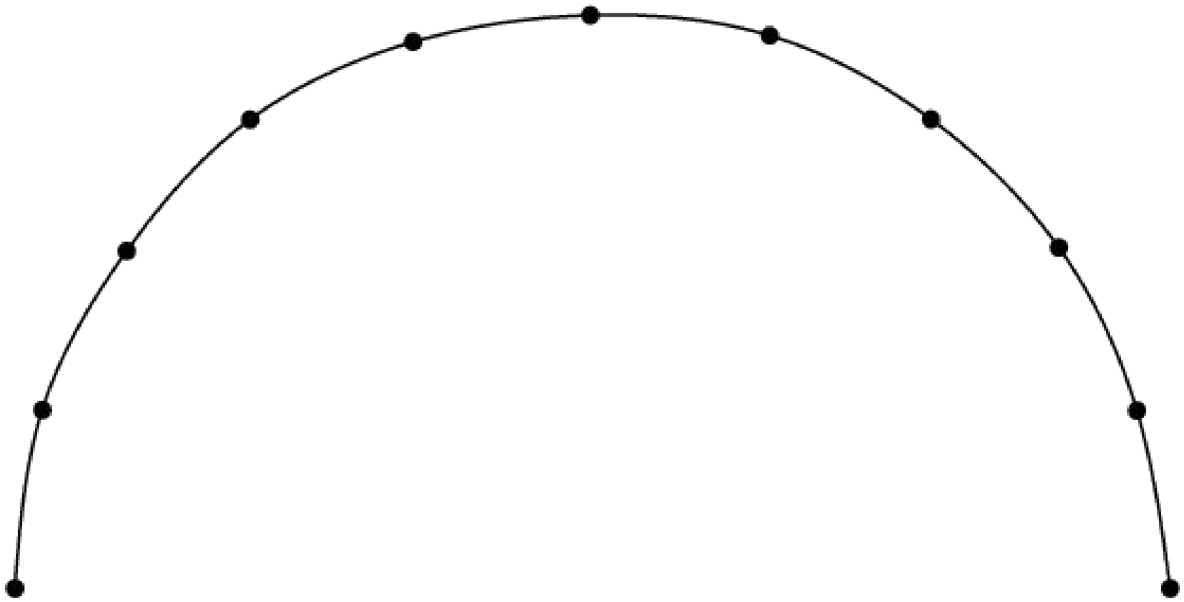
3 points, random perturbation 1%

Splines



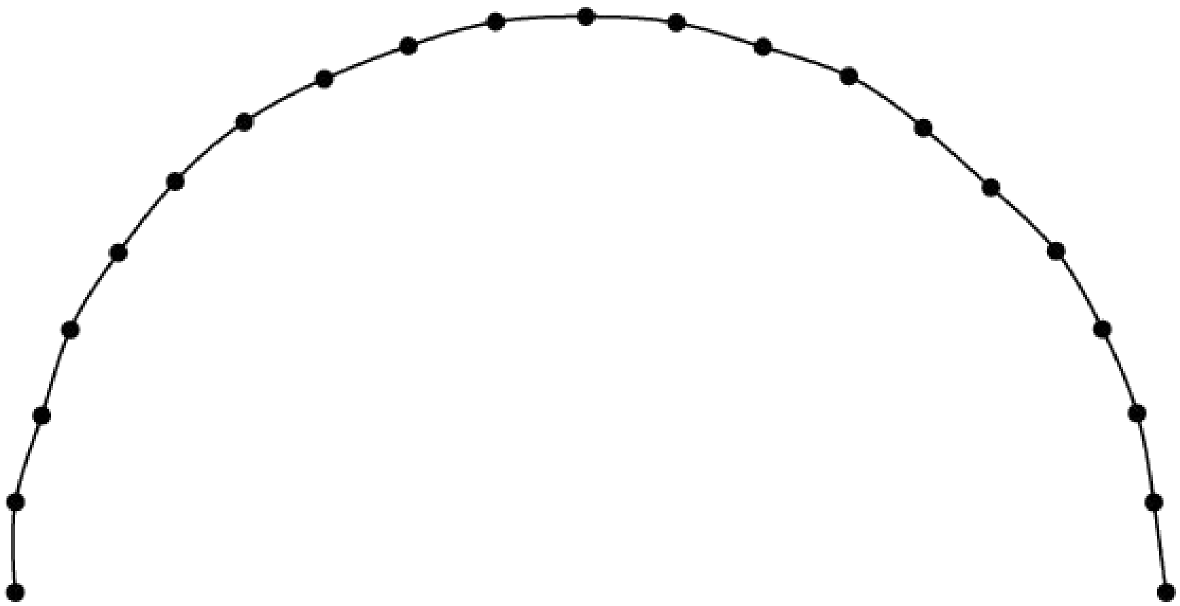
5 points, random perturbation 1%

Splines



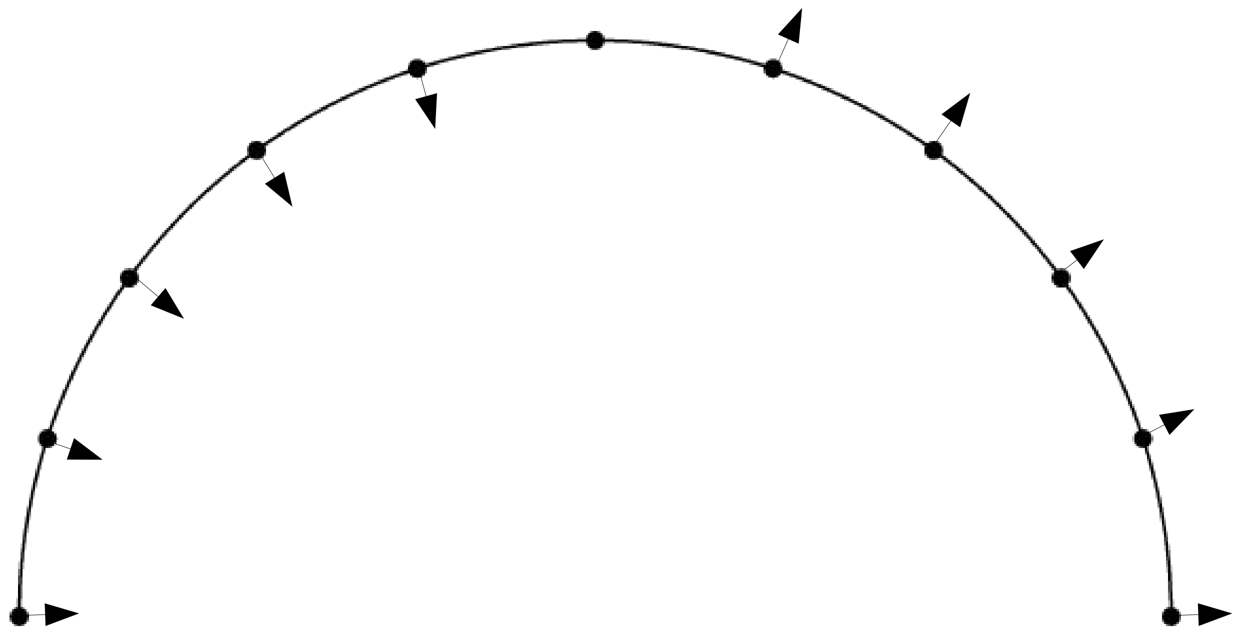
11 points, random perturbation 1%

Splines

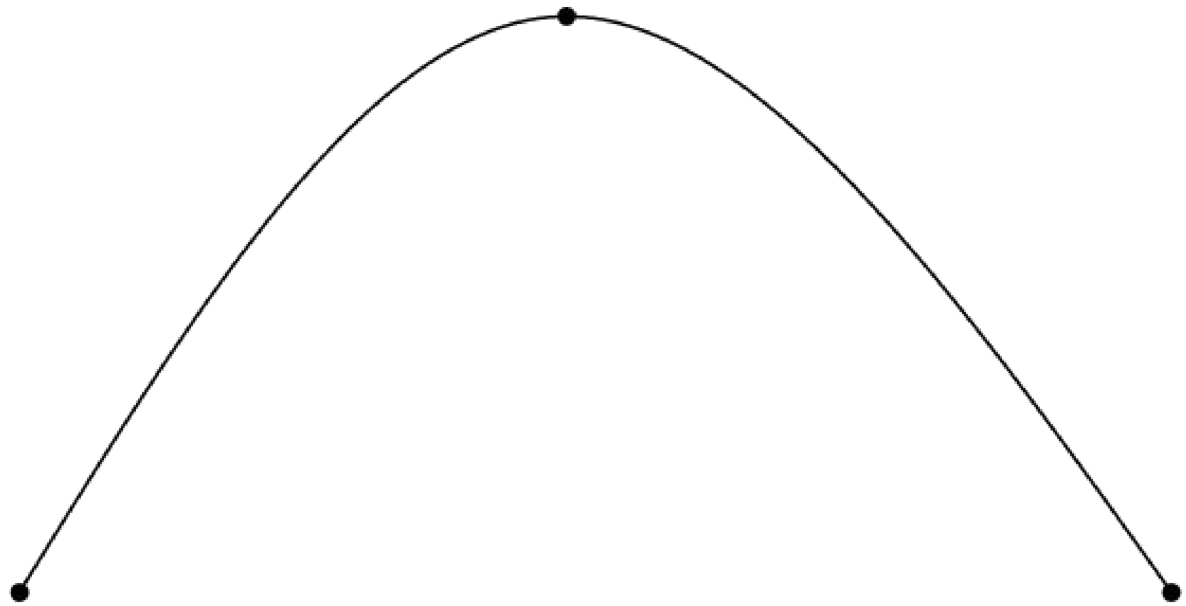


21 points, random perturbation 1%

- Deterministic perturbation
  - Each point is shifted radially depending on its position by -5 or +5 % of the circle's radius

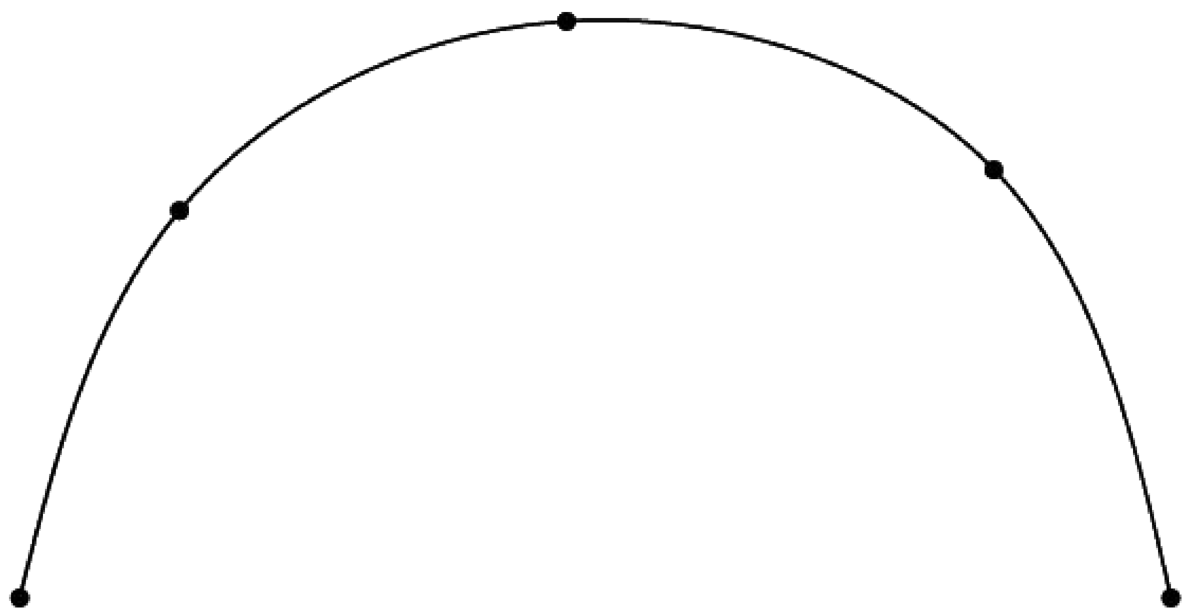


Splines



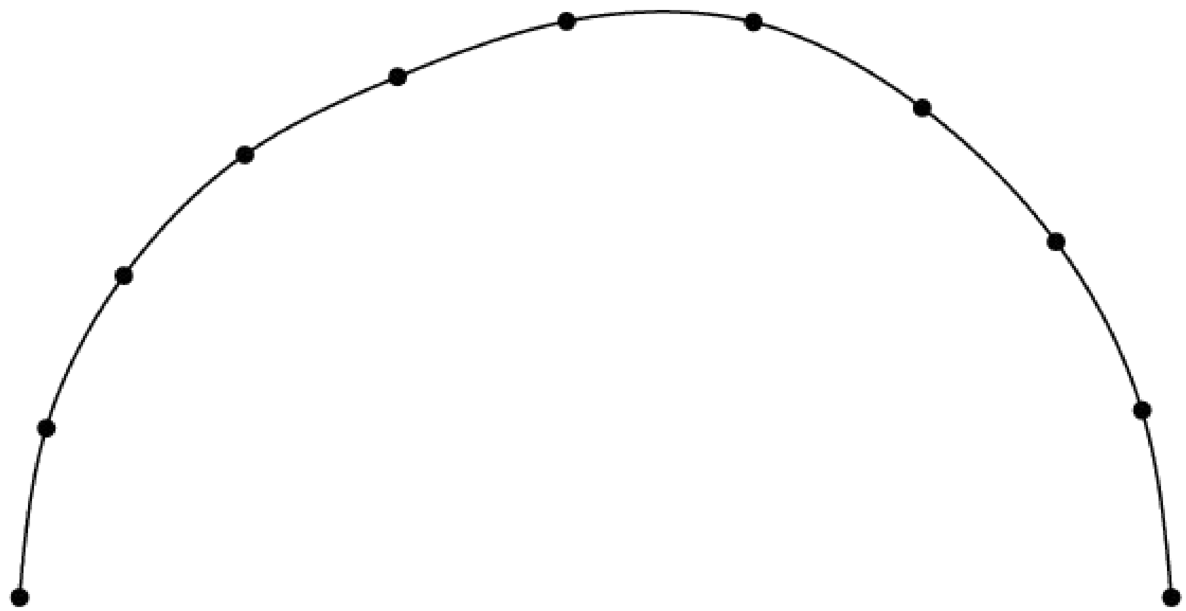
3 points, deterministic perturbation 5%

Splines



5 points, deterministic perturbation 5%

Splines

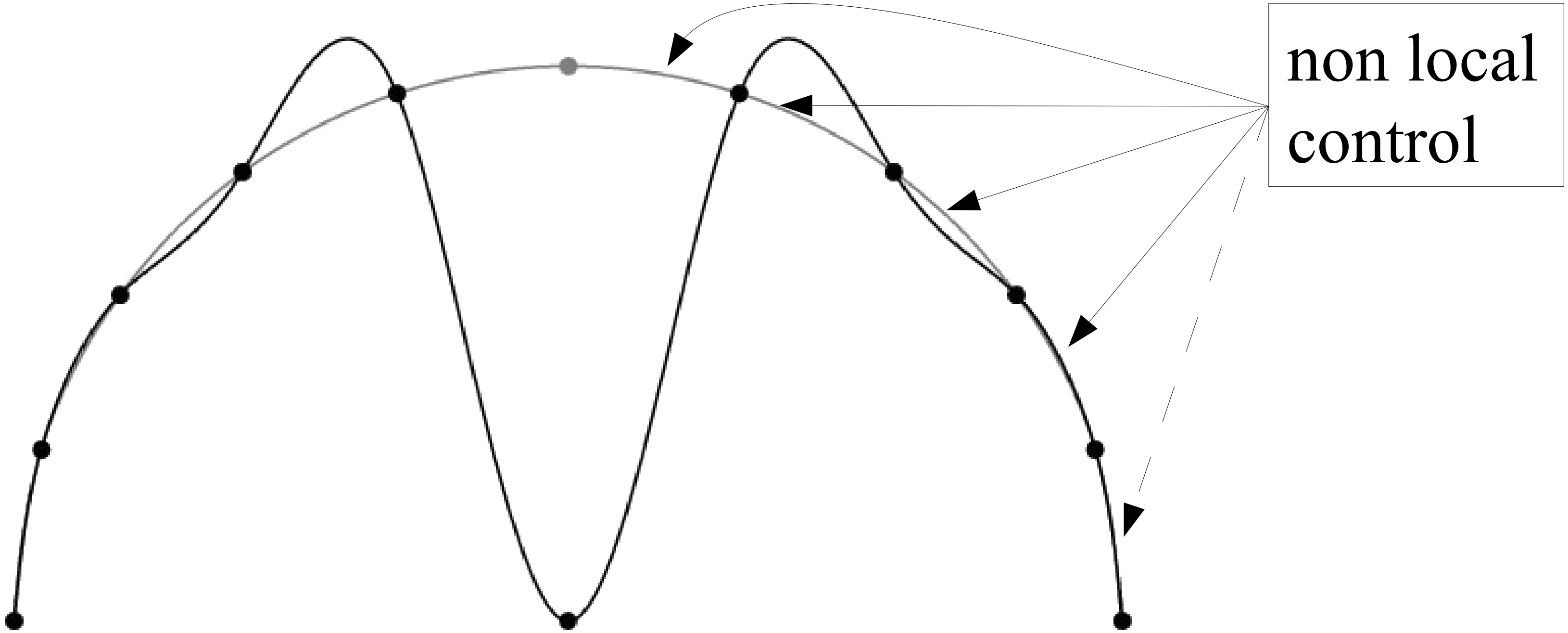


11 points, deterministic perturbation 5%

Splines

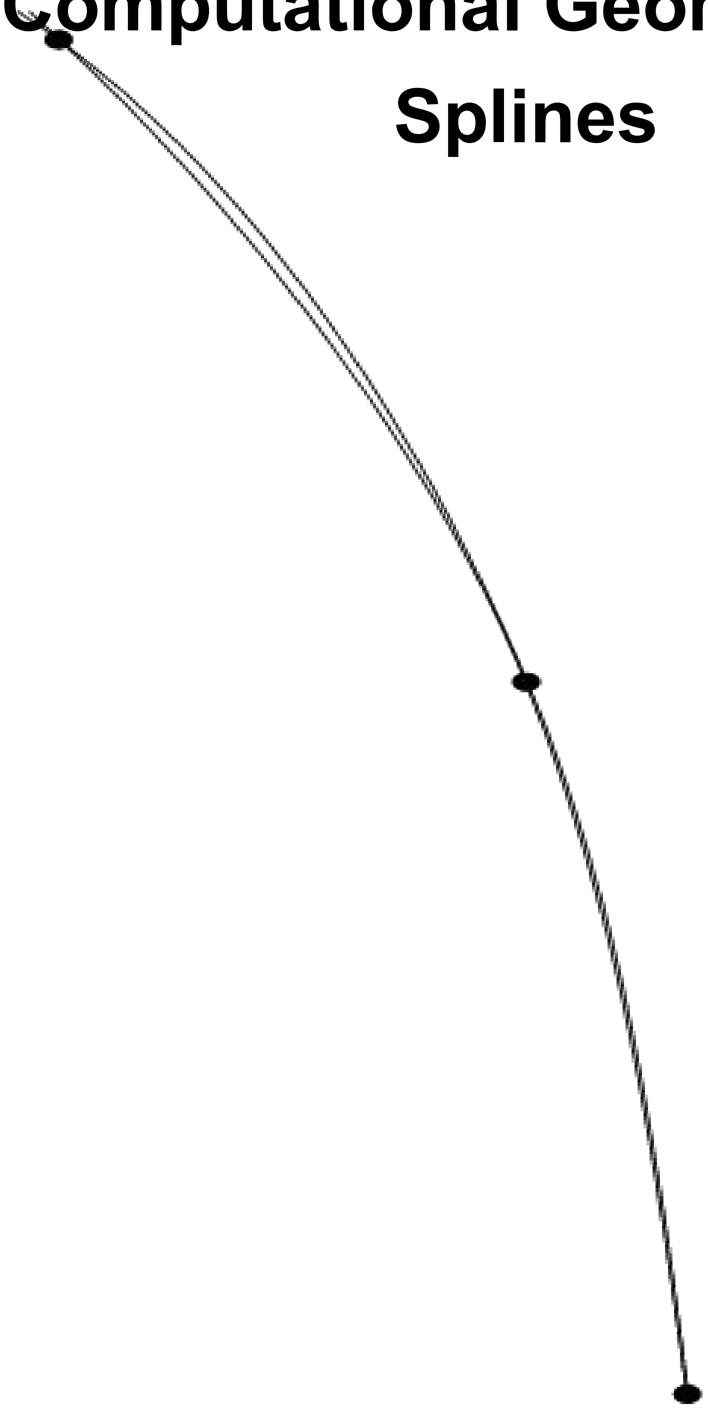
21 points, deterministic perturbation 5%

Splines

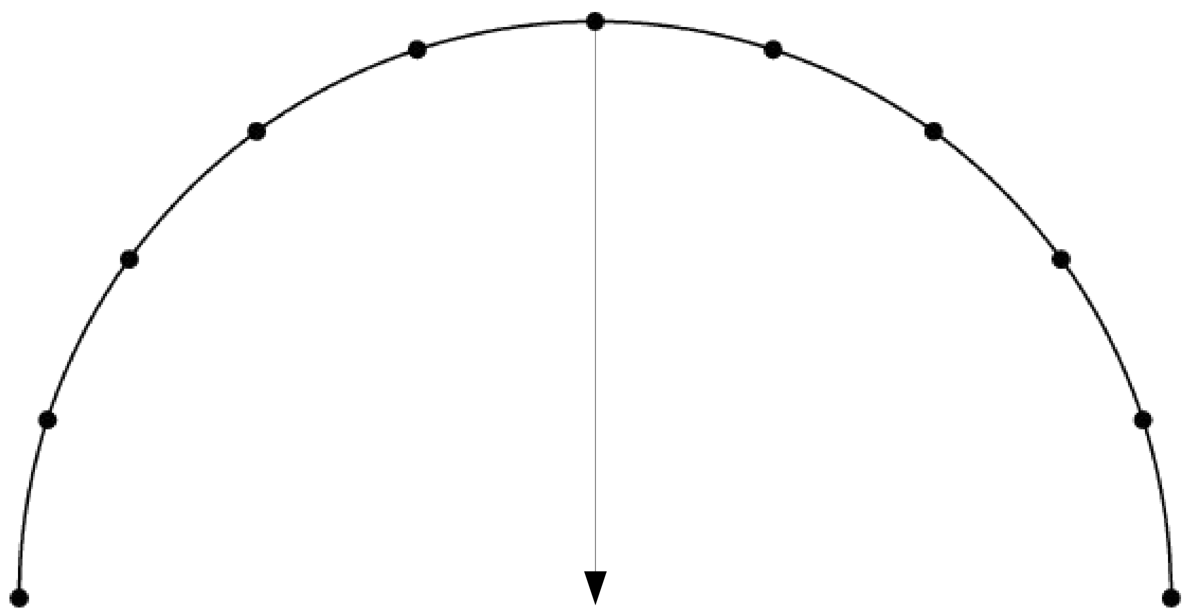


11 points

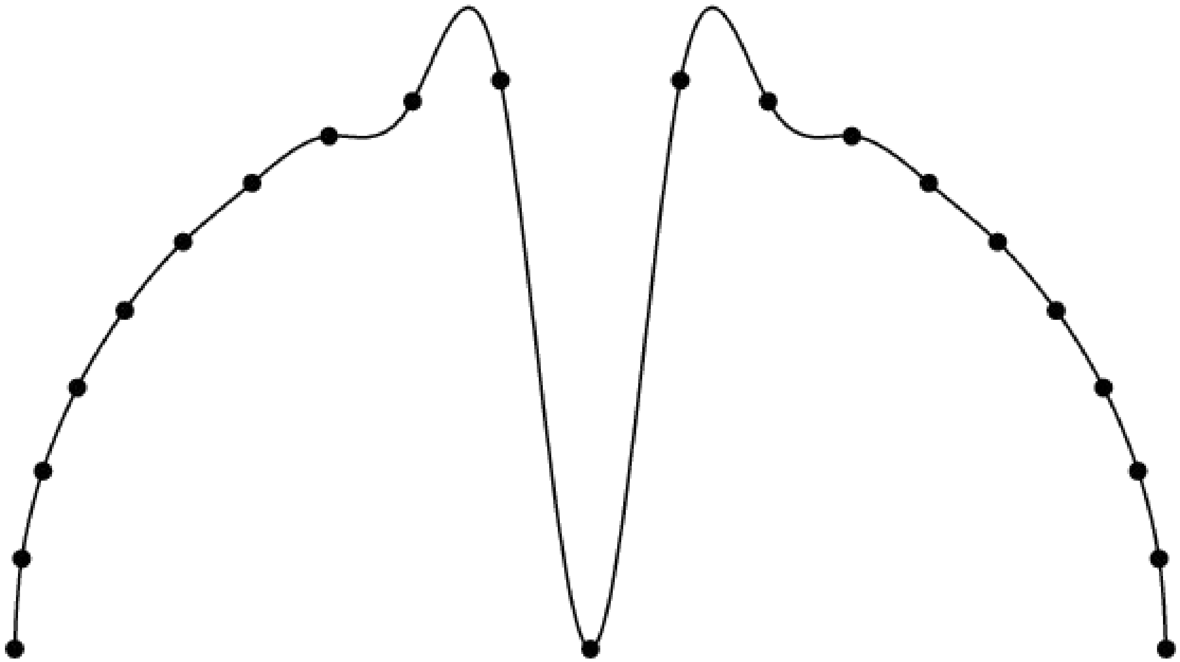
Splines



- Perturbation of a point
  - We shift one point by a significant amount

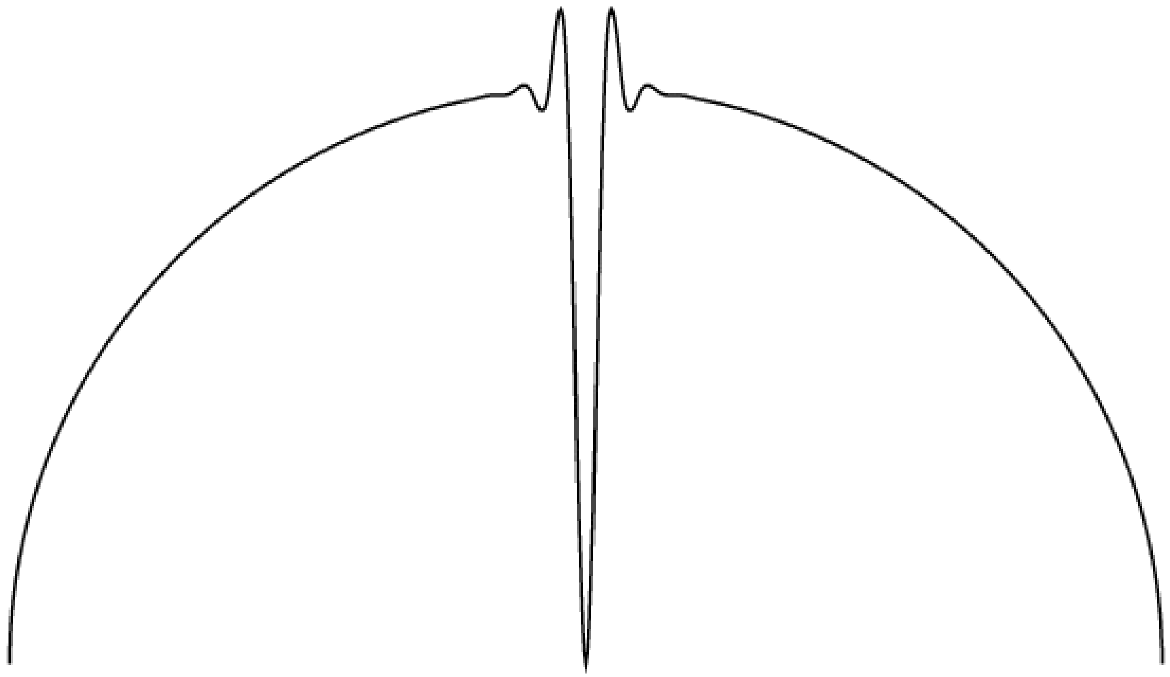


Splines



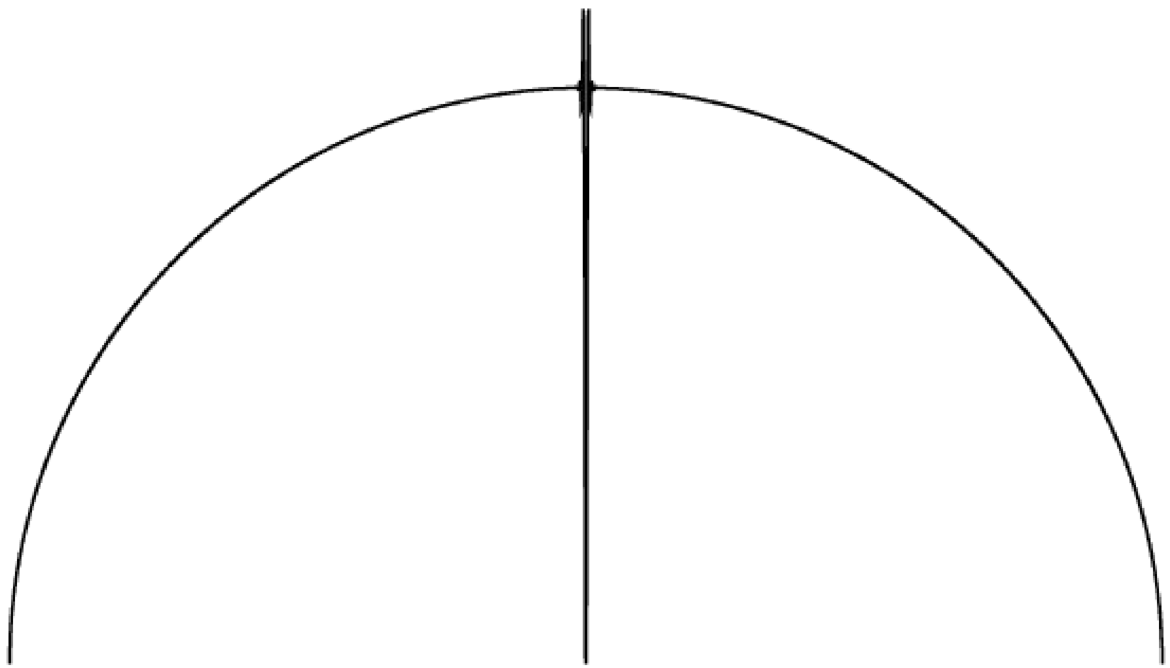
21 points

Splines



99 points

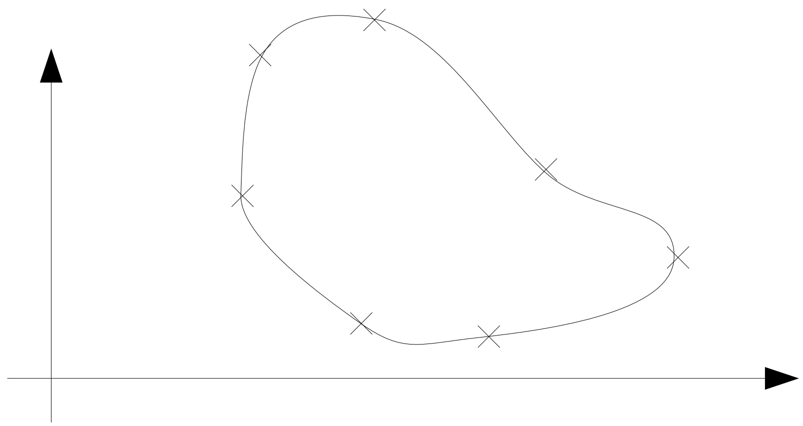
Splines



999 points

- Stable interpolation scheme
- Weak Runge phenomenon
- The displacement of a point yet affects all the curve
  - Nevertheless, the perturbation fades very quickly further away from the shifted point
  - « Overshoots » are limited.

- Closed curve ?
  - The curve can be closed, just impose everywhere that the second derivative is continuous.



- Instead of

$$x''_{[0]}(0)=0 \Leftrightarrow 2a_{[0]2}=0$$

$$x''_{[n-2]}(1)=0 \Leftrightarrow 2a_{[n-2]2}+6a_{[n-2]3}=0$$

... we have

$$x''_{[n-2]}(1)=x''_{[0]}(0) \Leftrightarrow 2a_{[n-2]2}+6a_{[n-2]3}=2a_{[0]2}$$

Circulant matrix

$$\begin{pmatrix} 4 & 1 & & & & 1 \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & & \ddots & & \\ & & & 1 & 4 & 1 \\ 1 & & & & 1 & 4 \end{pmatrix} \begin{pmatrix} x'_0 \\ x'_1 \\ x'_2 \\ \vdots \\ x'_{n-3} \\ x'_{n-2} \end{pmatrix} = \begin{pmatrix} 3(x_1-x_{n-2}) \\ 3(x_2-x_0) \\ 3(x_3-x_1) \\ \vdots \\ 3(x_{n-2}-x_{n-4}) \\ 3(x_0-x_{n-3}) \end{pmatrix}$$

and

$$x_{n-1}=x_0$$
$$x'_{n-1}=x'_0$$

- General case : arbitrary parametrization

$$P(u_i) = P_i \equiv \begin{cases} x(u_i) = x_i \\ y(u_i) = y_i \end{cases}$$

$$x_{[i]}(u) = A_{[i]0} + A_{[i]1} u + A_{[i]2} u^2 + A_{[i]3} u^3 \quad , \ u \in [u_i, u_{i+1}]$$

  - We again change the parametrization...

$$\bar{u} = \frac{u - u_i}{u_{i+1} - u_i} \qquad \frac{d\bar{u}}{du} = \frac{1}{u_{i+1} - u_i} = \frac{1}{h_i}$$

$$x_{[i]}(\bar{u}) = a_{[i]0} + a_{[i]1} \bar{u} + a_{[i]2} \bar{u}^2 + a_{[i]3} \bar{u}^3 \quad , \ \bar{u} \in [0,1]$$

- We pass through both points :

$$P(\bar{u}_0=0)=P_i \Leftrightarrow a_{[i]0}+a_{[i]1}\bar{u}_0+a_{[i]2}\bar{u}_0^2+a_{[i]3}\bar{u}_0^3=x_i$$

$$P(\bar{u}_1=1)=P_{i+1} \Leftrightarrow a_{[i]0}+a_{[i]1}\bar{u}_1+a_{[i]2}\bar{u}_1^2+a_{[i]3}\bar{u}_1^3=x_{i+1}$$

- We impose both slopes :

$$P'(\bar{u}_0=0)=\frac{dP}{du}(0)=\frac{dP}{d\bar{u}}(0)\frac{1}{h_i}=P'_i \Leftrightarrow a_{[i]1}+2\mathrm{a}_{[i]2}\bar{u}_0+3\mathrm{a}_{[i]3}\bar{u}_0^2=x_i' h_i$$

$$P'(\bar{u}_1=1)=\frac{dP}{du}(1)=\frac{dP}{d\bar{u}}(1)\frac{1}{h_i}=P'_{i+1} \Leftrightarrow a_{[i]1}+2\mathrm{a}_{[i]2}\bar{u}_1+3\mathrm{a}_{[i]3}\bar{u}_1^2=x_{i+1}' h_i$$

- Finally :

$$\begin{cases} a_{[i]0}=x_i \\ a_{[i]1}=x_i' h_i \\ a_{[i]2}=3(x_{i+1}-x_i)-2x_i' h_i-x_{i+1}' h_i \\ a_{[i]3}=2(x_i-x_{i+1})+x_i' h_i+x_{i+1}' h_i \end{cases}$$

Splines

- We impose the second derivative for a natural spline

$$\frac{d^2 P_{[i]}}{du^2}(\bar{u}) = \frac{d^2 P_{[i]}}{d\,\bar{u}^2}\frac{d^2\bar{u}}{d\,u^2} = \frac{d^2 P_{[i]}}{d\,\bar{u}^2}\frac{1}{h_i^{\,2}}$$

$$x''_{[i-1]}(1) = x''_{[i]}(0) \Leftrightarrow \frac{2\mathrm{a}_{[i-1]2} + 6\mathrm{a}_{[i-1]3}}{h_{i-1}^2} = \frac{2\mathrm{a}_{[i]2}}{h_i^2}$$

- We substitute in the internal equations

$$\frac{2[3(x_i - x_{i-1}) - 2x'_{i-1}h_{i-1} - x'_i h_{i-1}] + 6[2(x_{i-1} - x_i) + x'_{i-1}h_{i-1} + x'_i h_{i-1}]}{h_{i-1}^2}$$

$$= \frac{2[3(x_{i+1} - x_i) - 2x'_i h_i - x'_{i+1} h_i]}{h_i^2}$$

Splines

- We obtain finally :

$$\frac{x'_{i-1}+2x'_i}{h_{i-1}}+\frac{2x'_i+x'_{i+1}}{h_i}=3\frac{(x_i-x_{i-1})}{h_{i-1}^2}+3\frac{(x_{i+1}-x_i)}{h_i^2}$$

$$h_i(x'_{i-1}+2x'_i)+h_{i-1}(2x'_i+x'_{i+1})=3\frac{h_i}{h_{i-1}}(x_i-x_{i-1})+3\frac{h_{i-1}}{h_i}(x_{i+1}-x_i)$$

- At the boundaries we want a vanishing second derivative …

$$x''_{[0]}(0)=0 \Leftrightarrow \frac{2a_{[0]2}}{(h_0^2)}=0$$

$$x''_{[n-2]}(1)=0 \Leftrightarrow \frac{2a_{[n-2]2}+6a_{[n-2]3}}{(h_{n-2}^2)}=0$$

$$h_0(2x'_0+x'_1)=3(x_1-x_0)$$

$$h_{n-2}(x'_{n-2}+2x'_{n-1})=3(x_{n-1}-x_{n-2})$$

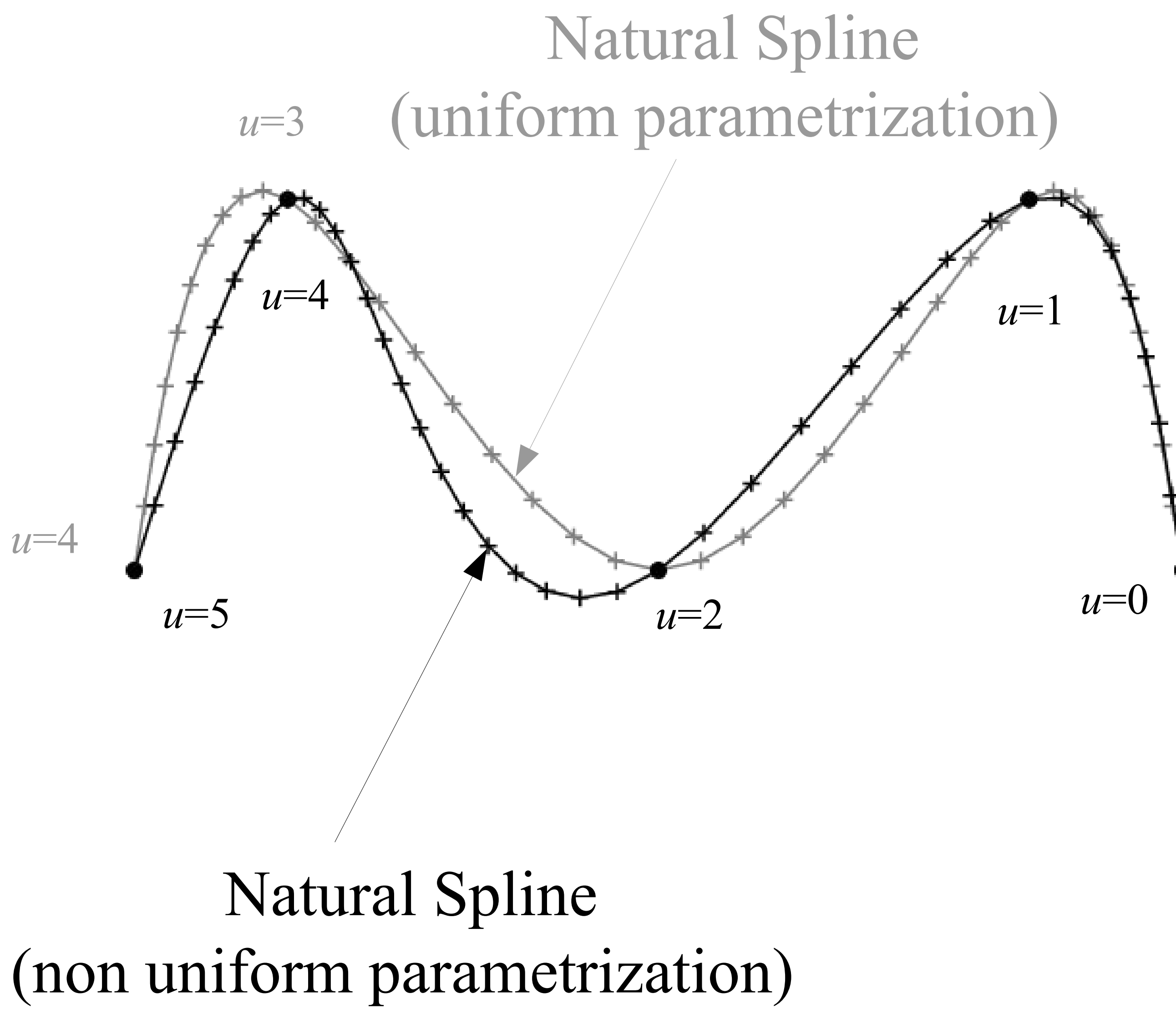
- We have then a linear system of n unknowns:

(next page)

Splines

$$
\begin{pmatrix}
2h_0 & h_0 & & & \\
h_1 & 2h_1+2h_0 & h_0 & & \\
 & & \ddots & & \\
 & & h_{n-2} & 2h_{n-2}+2h_{n-3} & h_{n-3} \\
 & & & h_{n-2} & 2h_{n-2}
\end{pmatrix}
\begin{pmatrix}
x_0' \\ x_1' \\ \vdots \\ x_{n-2}' \\ x_{n-1}'
\end{pmatrix}
=
\begin{pmatrix}
3(x_1-x_0) \\
3\frac{h_1}{h_0}(x_1-x_0)+3\frac{h_0}{h_1}(x_2-x_1) \\
\vdots \\
3\frac{h_{n-2}}{h_{n-3}}(x_{n-2}-x_{n-3})+3\frac{h_{n-3}}{h_{n-2}}(x_{n-1}-x_{n-2}) \\
3(x_{n-1}-x_{n-2})
\end{pmatrix}
$$

Splines


Natural Spline
(uniform parametrization)
u=3
u=4
u=4
u=1
u=5
u=2
u=0
Natural Spline
(non uniform parametrization)

- Compact notation
  - We would like a compact representation as for the Lagrange interpolation.

$$x_{[i]}(\bar{u}) = a_{[i]0} + a_{[i]1}\bar{u} + a_{[i]2}\bar{u}^2 + a_{[i]3}\bar{u}^3$$
$$0 \le \bar{u} < 1$$

$$\begin{cases} a_{[i]0} = x_i \\ a_{[i]1} = x_i' \\ a_{[i]2} = 3(x_{i+1} - x_i) - 2x_i' - x_{i+1}' \\ a_{[i]3} = 2(x_i - x_{i+1}) + x_i' + x_{i+1}' \end{cases}$$

$$\begin{cases} x_{[i]}(\bar{u}) = \sum_0^n x_i h_{i0}^p(\bar{u}) + \sum_0^n x_i' h_{i1}^p(\bar{u}) \\ y_{[i]}(\bar{u}) = \sum_0^n y_i h_{i0}^p(\bar{u}) + \sum_0^n y_i' h_{i1}^p(\bar{u}) \\ \quad\quad\quad \ldots \end{cases} \Leftrightarrow P_{[i]}(\bar{u}) = \sum_0^n P_i h_{i0}^p(\bar{u}) + \sum_0^n P_i' h_{i1}^p(\bar{u})$$

This on each interval $[i,i+1]$.

- We have on each interval :

$$\begin{cases} a_{[i]0} = x_i \\ a_{[i]1} = x_i' \\ a_{[i]2} = 3(x_{i+1} - x_i) - 2x_i' - x_{i+1}' \\ a_{[i]3} = 2(x_i - x_{i+1}) + x_i' + x_{i+1}' \end{cases}$$

$$x_{[i]}(\bar{u}) = x_i + x_i' \bar{u} + (3(x_{i+1} - x_i) - 2x_i' - x_{i+1}')\bar{u}^2 + (2(x_i - x_{i+1}) + x_i' + x_{i+1}')\bar{u}^3$$

- By rearranging equations

$$\begin{aligned} x_{[i]}(\bar{u}) = {} & x_i(1 - 3\bar{u}^2 + 2\bar{u}^3) + x_i'(\bar{u} - 2\bar{u}^2 + \bar{u}^3) \\ & + x_{i+1}(3\bar{u}^2 - 2\bar{u}^3) + x_{i+1}'(-\bar{u}^2 + \bar{u}^3) \end{aligned}$$

- 3D Curves
  - Minimal order so that a curve can have a torsion (non planar curve)
    - Let's consider a Lagrange interpolation $P(u)=\sum_{i=0}^{n-1} P_i l_i^p(u)$
    - 2 points → on a straight line (no curvature)
    - 3 points → in a plane (no torsion)
    - 4 points → torsion becomes possible
  - Minimal order to join smoothly two arbitrarily oriented curves = 3