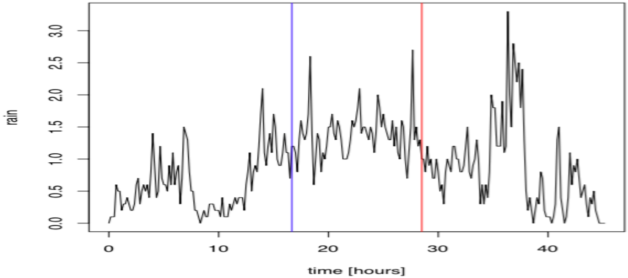
Numerics (Part IV)

[Lecture 0] Interpolation

Alessandro Antonucci
alessandro.antonucci@supsi.ch

Some examples

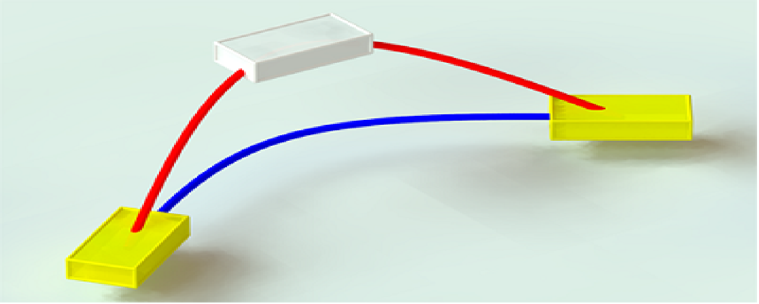
- Recording rainfall intensity every 10 minutes
- Intensity at 1:27pm?
- Total rainfall in a given time window



$$I(t = 13:27) = ? \qquad \int_{T_i}^{T_f} I(t)\mathrm{d}t = ?$$

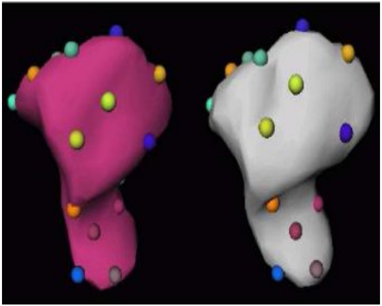
Some examples (ii)

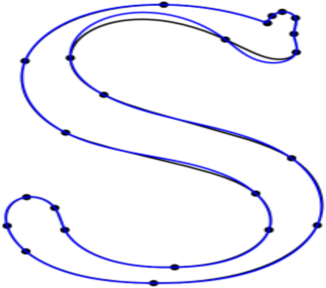
- Robot required to cross a number of points
- Curvature in trajectories has a miinimum
- Best path?



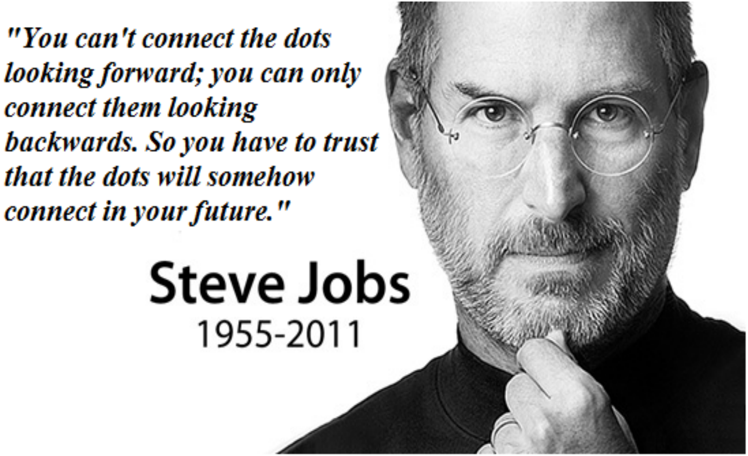
Some examples (iii)

- Design, animation (2D, 3D), . . .

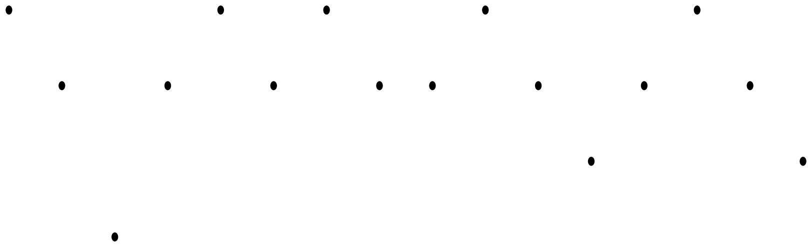




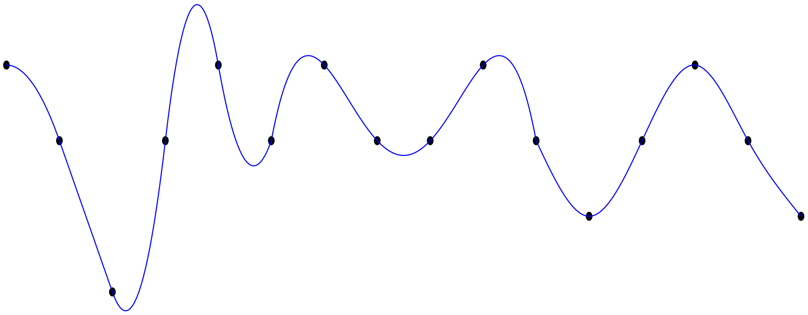
Connecting the dots



Connecting the dots



Connecting the dots



Interpolation Algorithms

Input

- Given a set of $n+1$ points $\{(x_i, y_i)\}_{i=0}^{n}$
- Such that $x_{i+1} > x_i \ \forall i$ (from left to right, no same x)

Output

- Finding function $f(x)$ defined between x_0 and x_n
- Such that $f(x_i) = y_i \ \forall i = 0, 1, \ldots, n$ (f touches the points)
- continuous f (optionally also derivatives continuous)

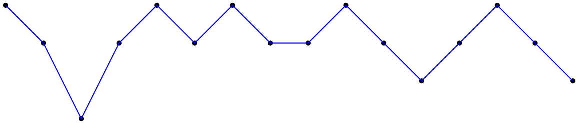
Different algorithms produce different interpolations
in general cannot say which one is better

The simplest interpolation algorithm

- Linear sp-line: consecutive pairs of points connected by straight lines
- Pros: easy to implement

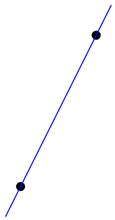
$$f(x) = y_j + \frac{y_{j+1} - y_j}{x_{j+1} - x_j}(x - x_j) \quad \text{if} \quad x \in [x_j, x_{j+1}]$$

- Cons: discontinuous derivative (sharp points)



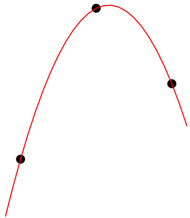
Polynomial interpolation

- 2 points ? Connected by one (and only one) straight line (polynomial of degree 1)



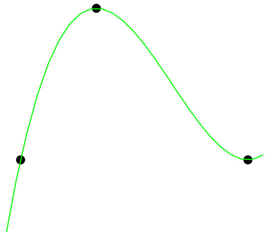
Polynomial interpolation

- 2 points ? Connected by one (and only one) straight line (polynomial of degree 1)
- 3 points ? Connected by one (and only one) parabolic curve (polynomial of degree 2)



Polynomial interpolation

- 2 points ? Connected by one (and only one) straight line (polynomial of degree 1)
- 3 points ? Connected by one (and only one) parabolic curve (polynomial of degree 2)
- 4 points ? Connected by one (and only one) cubic curve (polynomial of degree 3)
- ...
- $n + 1$ points ? Connected by one (and only one) function of degree n
- Pros: infinitely many derivatives



Algorithm for Polynomial Interpolation

- Generic n-degree polynomial : $p_n(x) = \sum_{i=0}^{n} a_i x^i$
- $n+1$ (unknown) parameters
- Requiring the $n+1$ points to be crossed linear system $(n+1) \times (n+1)$:

$$
\begin{array}{ccccc|c}
a_0 & a_1 & a_2 & \dots & a_n & \\
\hline
1 & x_0 & x_0^2 & \dots & x_0^n & y_0 \\
1 & x_1 & x_1^2 & \dots & x_1^n & y_1 \\
 & & \dots & \dots & & \\
1 & x_n & x_n^2 & \dots & x_n^n & y_n
\end{array}
$$

- Cubic algorithm for polynomial interpolation

A better algorithm for polynomial interpolation

- Alternative formulation for polynomials, ex. $p_3(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + c_3(x - x_0)(x - x_1)(x - x_2)$
- $p_n(x) = \sum_{i=0}^{n} c_i \left[\prod_{j=0}^{i-1}(x - x_j)\right]$
- Forcing the polynomial to cross the $n+1$ points

$$
\left[\begin{array}{ccccc|c}
1 & 0 & 0 & \dots & 0 & y_0 \\
1 & x_1 - x_0 & 0 & \dots & 0 & y_1 \\
 & & \dots & \dots & & \\
1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & \dots & \prod_{j=0}^{n-1}(x_n - x_j) & y_n
\end{array}\right]
$$

- Triangular system (after swaps), $O(n^2)$ solution!