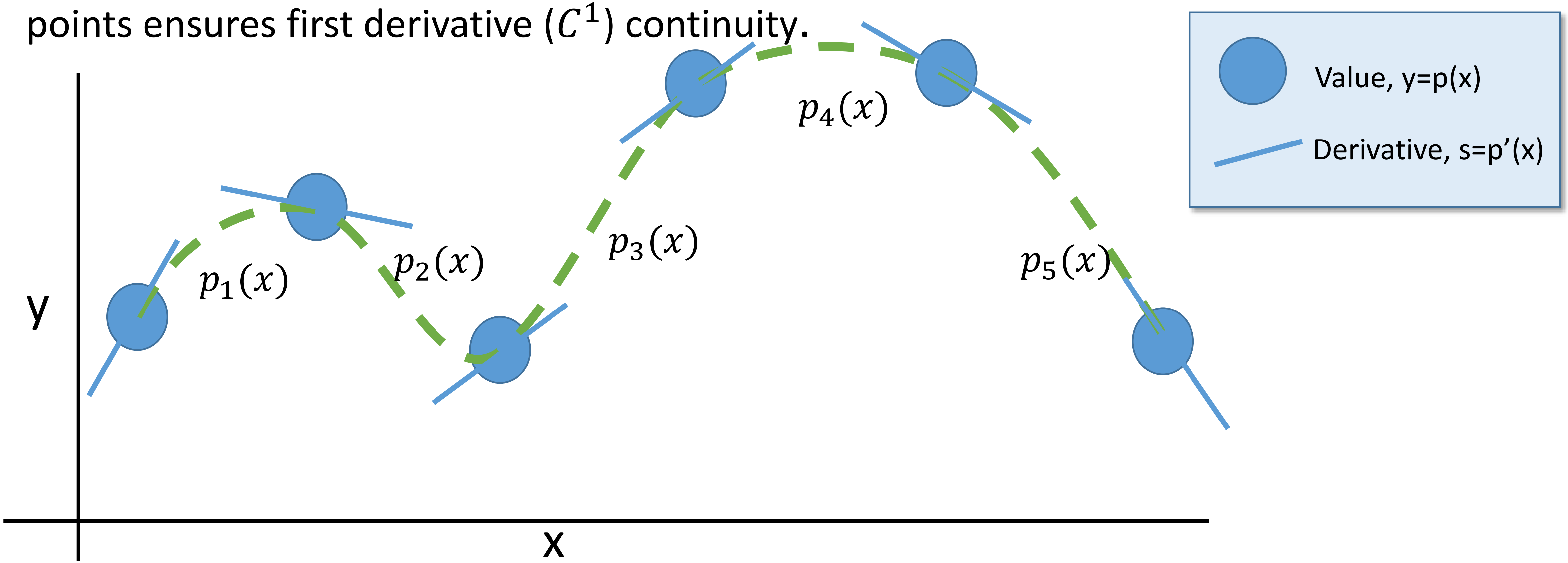
Interpolation – Cubic Splines

CS370 – May 13, 2016

Hermite interpolation – Many points

Fit many points (given values **and** 1st deriv.) with ***piecewise*** Hermite interpolation?

Use ***one cubic per pair*** of (adjacent) points. The matching/shared derivative data at points ensures first derivative (C^1) continuity.



Hermite interpolation – General solution

If we (instead) define the polynomial on the i^{th} interval, $p_i(x)$, as

$$p_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

there exist *direct* formulas for the polynomial coefficients:

Spacing of points in x.

$$a_i = y_i$$

$$\Delta x_i = x_{i+1} - x_i$$

$$b_i = s_i$$

$$c_i = \frac{3y_i' - 2s_i - s_{i+1}}{\Delta x_i}$$

where we define

and

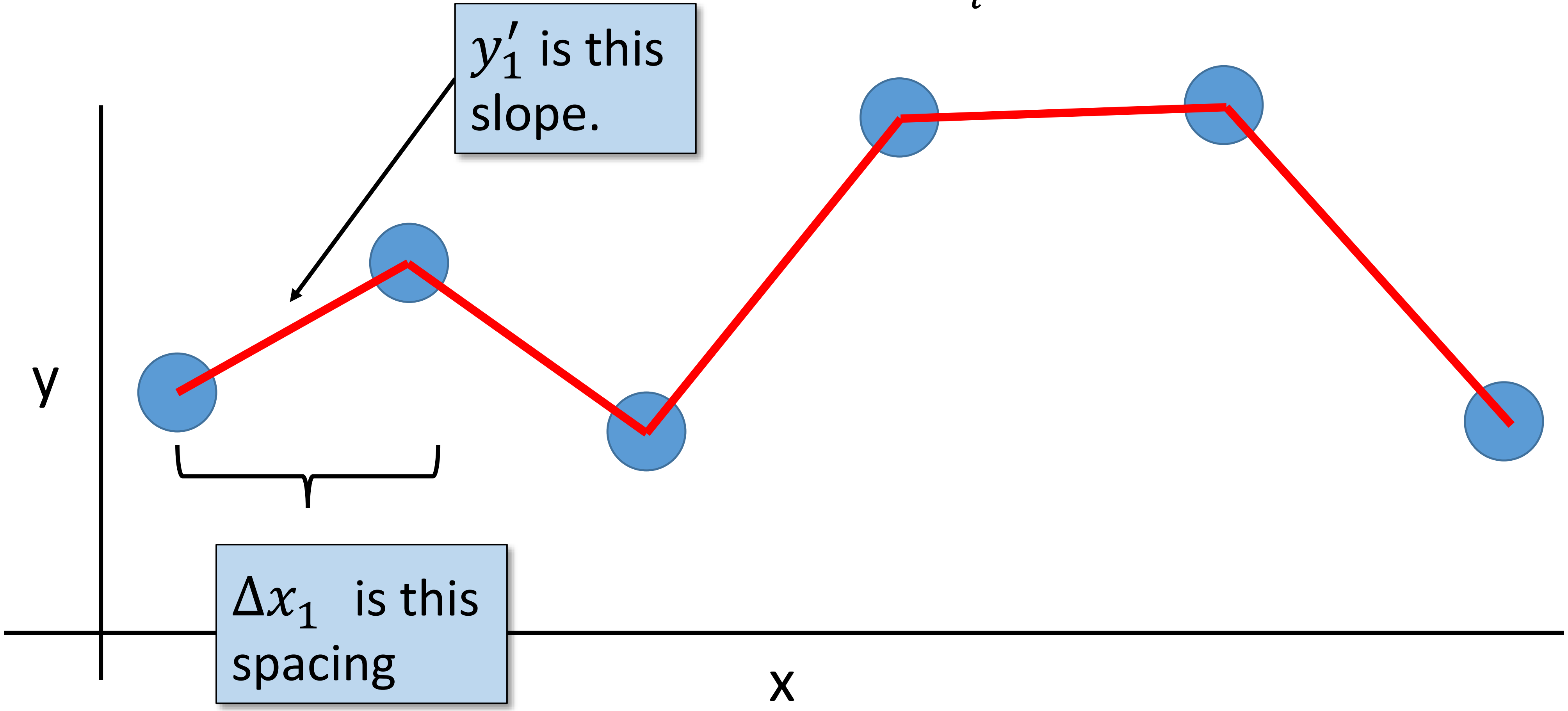
$$d_i = \frac{s_{i+1} + s_i - 2y_i'}{\Delta x_i^2}$$

$$y_i' = \frac{y_{i+1} - y_i}{\Delta x_i}$$

Slopes for a piece-wise *linear* fit.

Hermite interpolation – General solution

We defined $\Delta x_i = x_{i+1} - x_i$ and $y_i' = \frac{y_{i+1} - y_i}{\Delta x_i}$.



Hermite example

Same example as before but using the direct form (for just one interval).

Hermite interpolant direct form:

$$p_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

with

$a_i = y_i$

$b_i = s_i$

$c_i = \frac{3y_i' - 2s_i - s_{i+1}}{\Delta x_i}$

$d_i = \frac{s_{i+1} + s_i - 2y_i'}{\Delta x_i^2}$

$\Delta x_i = x_{i+1} - x_i$

$y_i' = \frac{y_{i+1} - y_i}{\Delta x_i}$

Consider two points: $p(0) = 0, p'(0) = 1, p(1) = 3, p'(1) = 0.$

Find the polynomial using the closed form above.

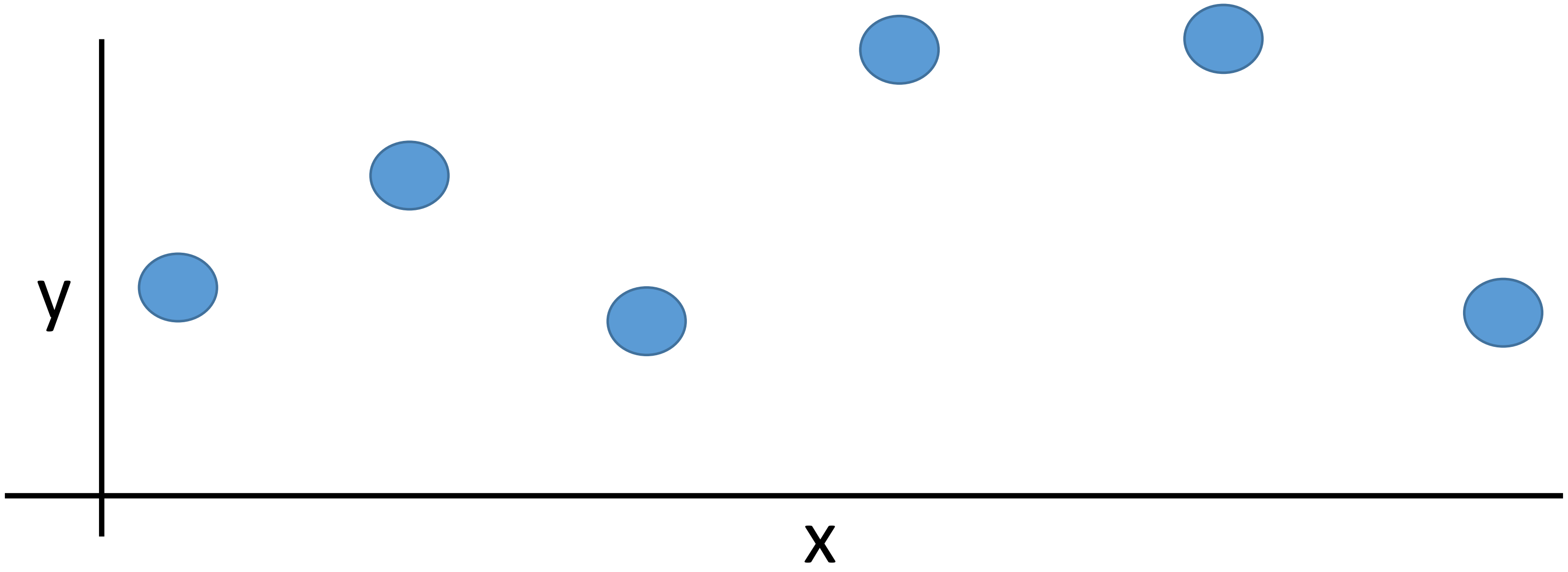
Under this notation, our input data is:

$$x_1 = 0, y_1 = 0, s_1 = 1, x_2 = 1, y_2 = 3, s_2 = 0.$$

Piecewise cubic (spline) interpolation

More common setting: ***no derivative information*** is given, just points.

Can we still fit a piecewise cubic to the set of points?



Yes, but each "piece" needs data from more than just its 2 endpoints…

The intervals cannot be computed independently!

Physical (Drafting) Splines

spline (n.)
long, thin piece of wood or metal, 1756, from East Anglian dialect, of uncertain origin.

Drafting “whales” or “ducks”

Cubic Splines – Main Idea

Approach:

Fit a cubic S_i on each sub-interval, but now require matching first *and* second derivatives (i.e., C^2 continuity).

Require "interpolating conditions" on each interval $[x_i, x_{i+1}]$,

$$S_i(x_i) = y_i, \qquad S_i(x_{i+1}) = y_{i+1}$$

Interval endpoint values match.

and "derivative conditions" at each *interior* point

$$S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}),$$
$$S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$$

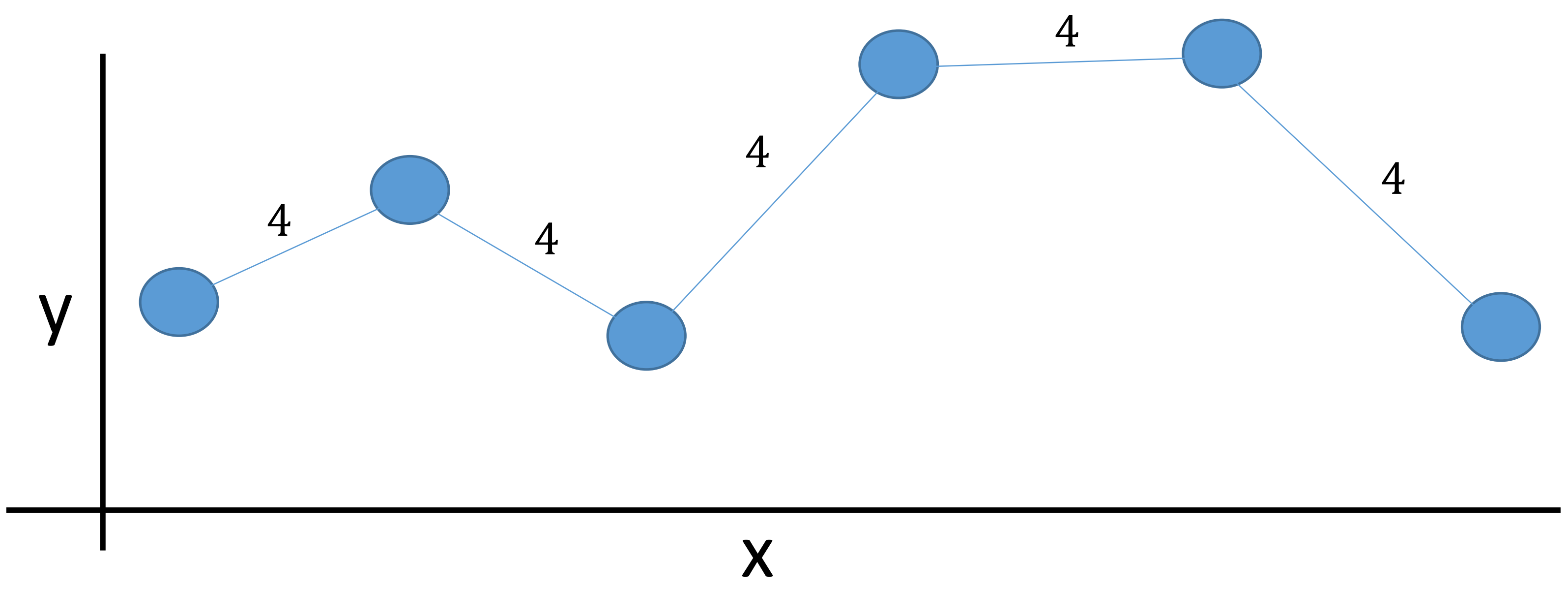
Interior 1st and 2nd derivatives match.

Note subscripts on S!

Counting Unknowns

Assuming n data points, how many *unknowns* do we have?

A cubic (4 unknowns) for each of $n - 1$ intervals, so $\boxed{\mathbf{4n - 4 \text{ unknowns}}}$.



e.g.,
$n = 6$ points,
$n - 1 = 5$ intervals,
$\therefore 4 \times 5 = 20$ unknowns.

Counting Equations

Assuming n data points, how many *equations* do we have?

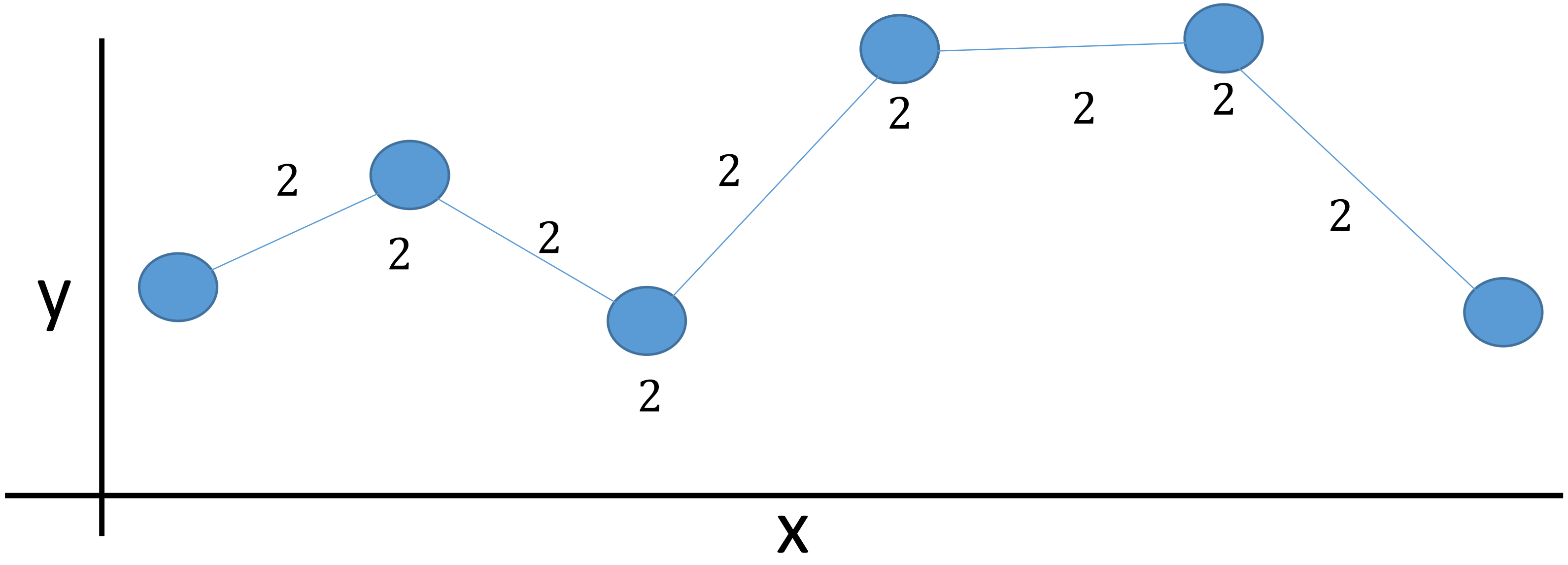
2 interpolating conditions per *interval*: $2(n-1) = 2n-2$

$$S_i(x_i) = y_i, \qquad S_i(x_{i+1}) = y_{i+1}$$

2 derivative conditions per *interior point*: $2(n-2) = 2n-4$

$$S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}), \qquad S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$$

$\boldsymbol{4n - 6}$ **equations.**



e.g.,
$2(5)$ interp. conditions
$+2(4)$ interior points
$= 18$ equations.

Solve the system?

$4n - 6$ equations, $4n - 4$ unknowns; can we solve the system?

No, $4n - 6 < 4n - 4$. Not enough equations to solve for all the coefficients!

Need 2 more constraints at the *domain endpoints*, called **boundary conditions** or **end conditions**.

Several choices exist…

Boundary conditions – Free/natural/variational

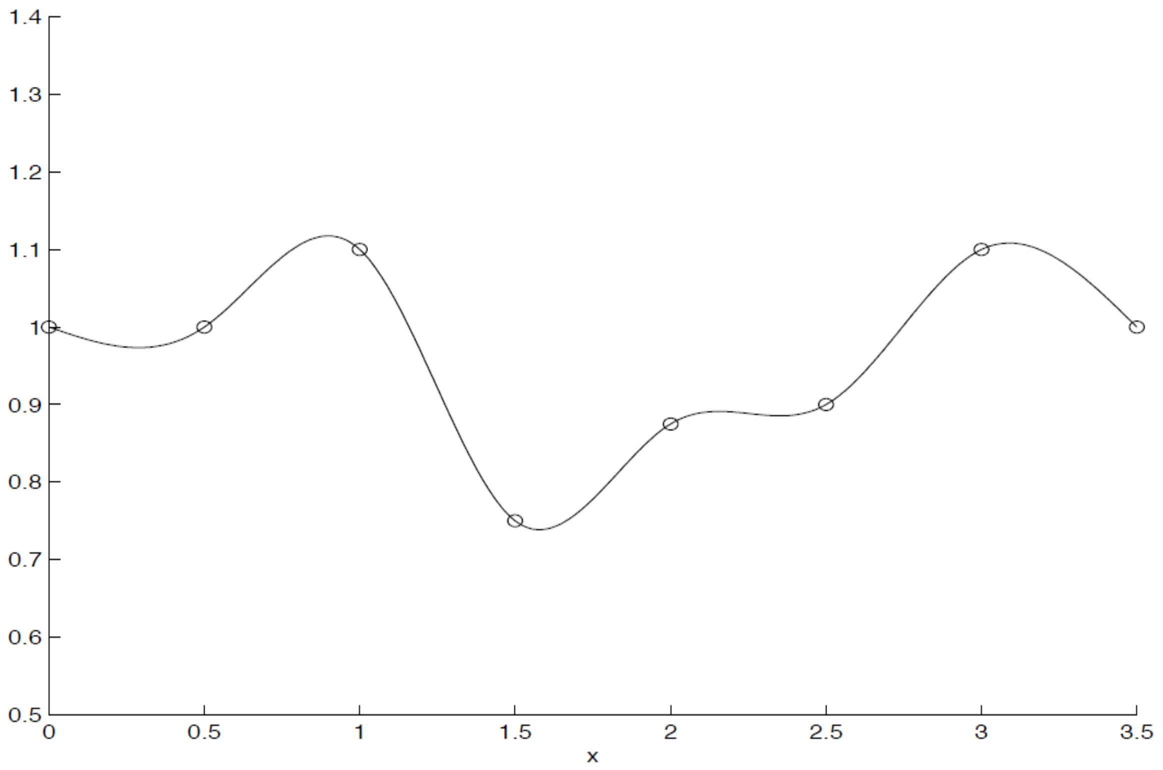
Free boundary condition: Second derivatives set to zero.

$$S''(x_1) = 0, \qquad S''(x_n) = 0$$

If both boundaries are free, called a *natural cubic spline*.

"Curvature" goes to zero at the end points, so the curve "straightens out".

(Technically not true geometric curvature, but does measure roughly how curved the function is.)



Boundary conditions – Clamped/complete

Clamped boundary conditions: Slope is set to specified value.

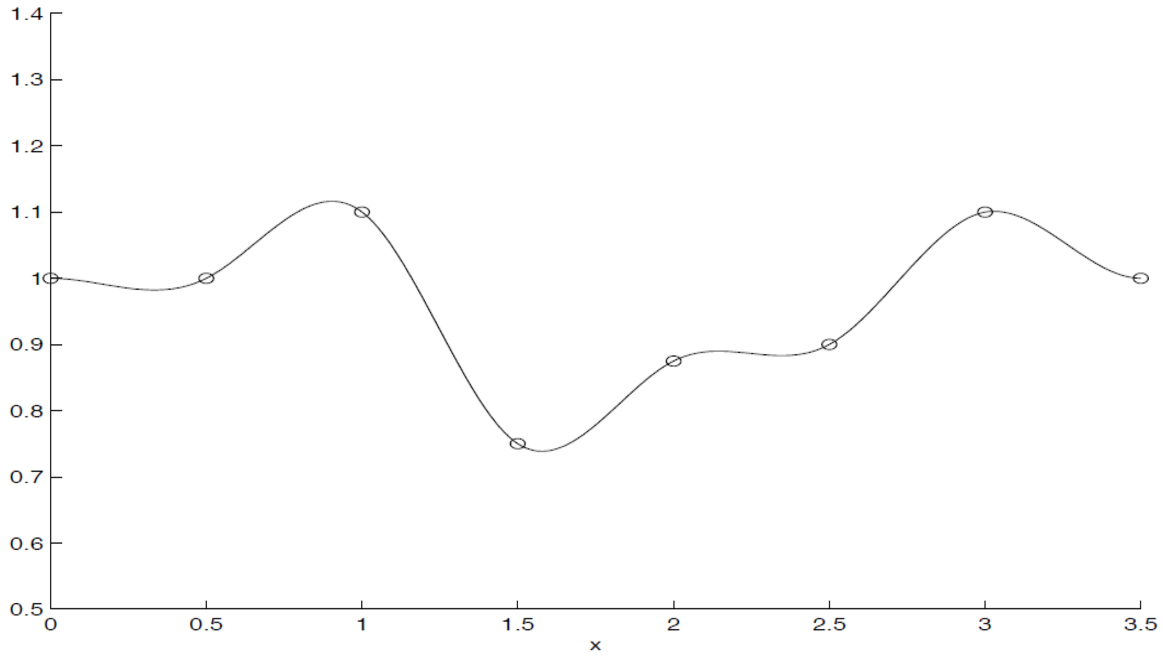
$S'(x_1) = \textit{specified}, S'(x_n) = \textit{specified}.$

If both boundaries *clamped*, it is a *complete* or *clamped spline*.

Here, $S'(x_1) = S'(x_n) = 0.$

i.e., the slope is prescribed to be zero,

so the curve ends become **horizontal**.



Aside: Boundary conditions & physical splines

The names “free” and “clamped” BC come from the analogy with physical splines.

Clamped BC are analogous to physically clamping the end of the rod in a particular direction.

The free condition is what you get if you don’t constrain the end. It is “free” to take on the most relaxed (uncurved) state.

Other boundary condition choices

- Periodic boundary conditions:
  - $S_1'(x) = S_n'(x), S_1''(x) = S_n''(x)$.
  - Endpoint derivatives match each other.
  - Gives a “wrap-around” behaviour.

- “Not-a-knot” boundary conditions:
  - $S_1'''(x) = S_2'''(x), S_{n-1}'''(x) = S_n'''(x)$.
  - Last two segments on the end become the *same polynomial*.
  - Since there’s no “switch” between polys, it’s “not a knot”.

Hermite interpolant v.s. cubic splines

How do the linear systems differ in size between Hermite case (given values+slopes), and cubic spline case (given only values)?

Hermite interpolation – each interpolant can be found independently.

- Solve $n - 1$ independent systems of size 4×4 (i.e. small).

Cubic spline – must solve for all polynomials together at once!

- Solve one large system of size $4(n - 1)$.

Physical Splines & Energy

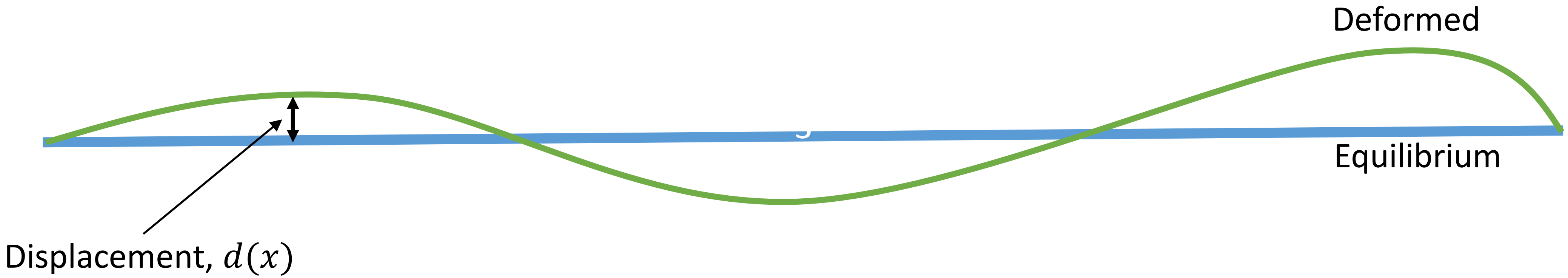
spline (n.)
long, thin piece of wood or metal, 1756, from East Anglian dialect, of uncertain origin.

Drafting "whales" or "ducks"

Spline Energy

There is an analogy between physical splines and cubic splines.

Bending a spline by placing the “drafting ducks” introduces some stored potential energy (the *bending* energy).



Real Spline Energy

What can we say about the shape that a physical spline will take on?

It finds the minimum (potential) energy configuration.
i.e., “least bent” smooth shape.

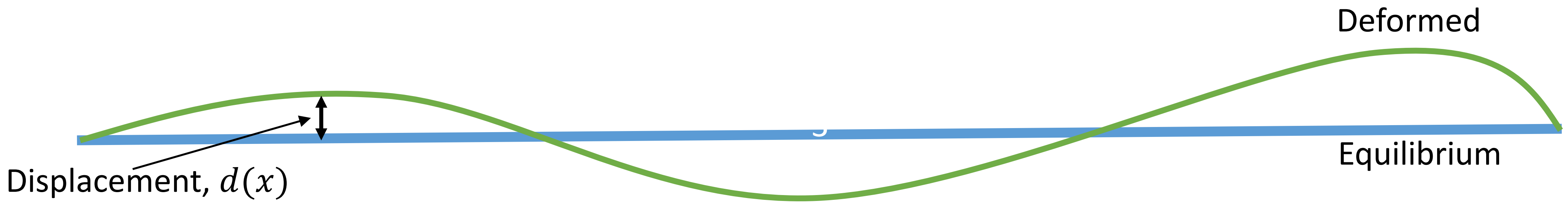
Our mathematical cubic splines likewise minimize an “energy”.

Cubic Spline Energy

If $d(x)$ is the displacement of the curve from flat state, cubic splines minimize the integral ("energy")

$$\int d''(x)^2$$

over all possible functions matching the interpolation points.



[See section 2.6 of the course notes for proof that the resulting curve has the lowest energy.]

Outcome – Are we actually better off?

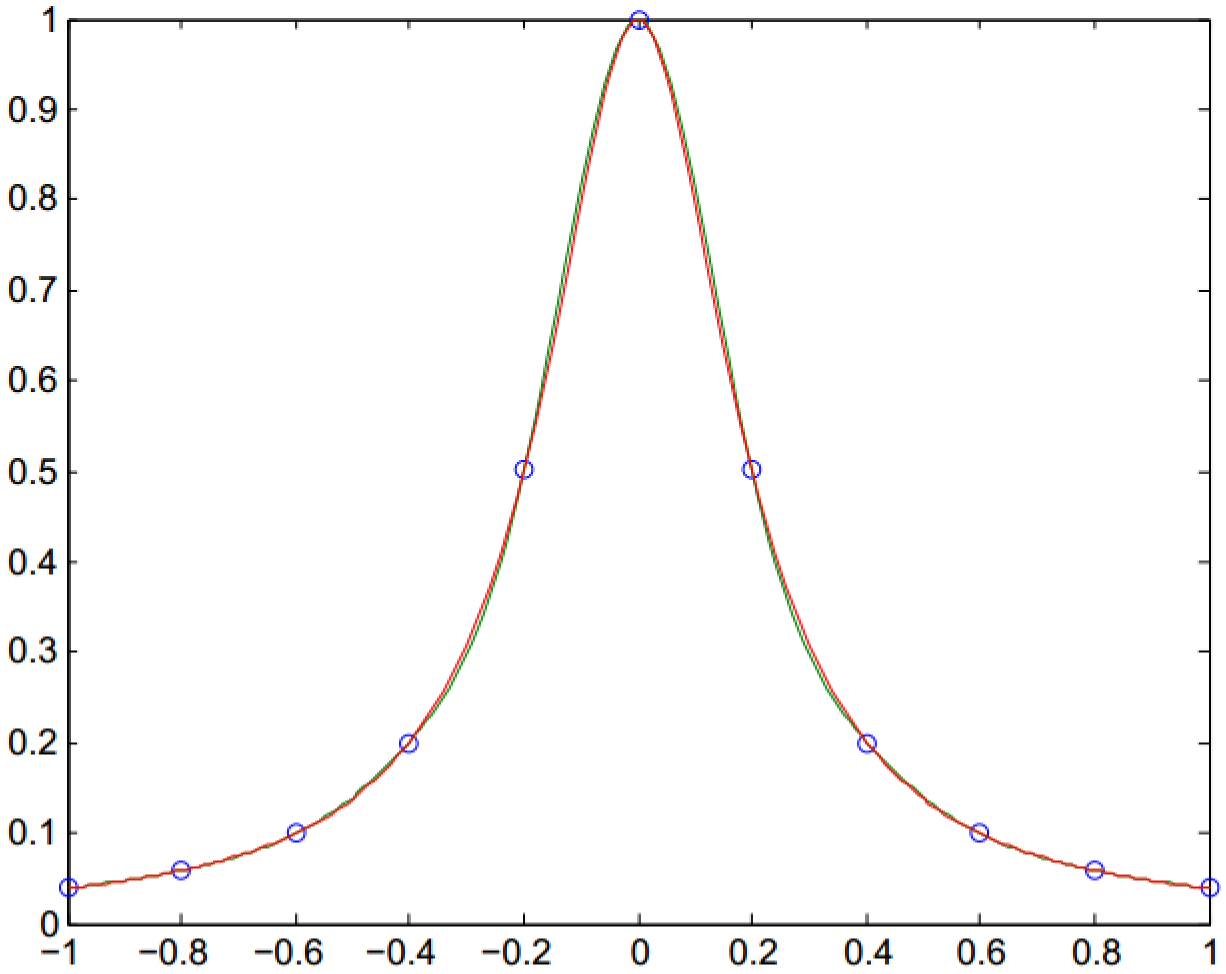
A cubic spline for the same (Runge) function from before,

$$f(x) = \frac{1}{1 + 25x^2}$$

fit to the same 11 points.

Visually, the curves are nearly superimposed.

We've successfully avoided the extreme oscillations from before!



Recap – Piecewise Polynomial Interpolation

- To circumvent “Runge’s phenomenon” (large oscillation for high degree polynomial fits to many points), we used *piecewise polynomials*.

- *Hermite interpolation* lets us fit piecewise cubics to data, given function values *and* derivatives.

- *Cubic spline interpolation* lets us fit piecewise cubics such that 1st and 2nd derivatives are continuous, given only the *values*.

Next Time: Smarter Solution Procedure?

So far, we have a linear system with $4n - 4$ equations for n points,

i.e., $Ax = b$.

We'll see later that solving general linear systems has complexity $O(N^3)$ for N unknowns (e.g. Gaussian elimination.)

Next class, we'll consider a more efficient approach to find a solution for cubic splines, *using Hermite interpolation.*