Engineering 7: Introduction to computer programming for scientists and engineers

Interpolation

- Recap
- Polynomial interpolation
- Spline interpolation

Regression and Interpolation:
“learning” functions from data

Regression and Interpolation:

Training Data: $(x_i, y_i)_{i=1,\cdots,N}$

Regression

Find a function, from a specified class, that best fits the data. For example,

$$\min_{f \in \mathbf{F}_R} \sum_{i=1}^{N} |f(x_i) - y_i|^2$$

Often use a squared-error criterion to score a function's performance.

Let f^R denote the best-fit function.

Interpolation

Find a function, from a specified class, that exactly matches the data. For example, find $f \in \mathbf{F}_I$ with

$$f(x_i) = y_i \text{ for all } 1 \leq i \leq N$$

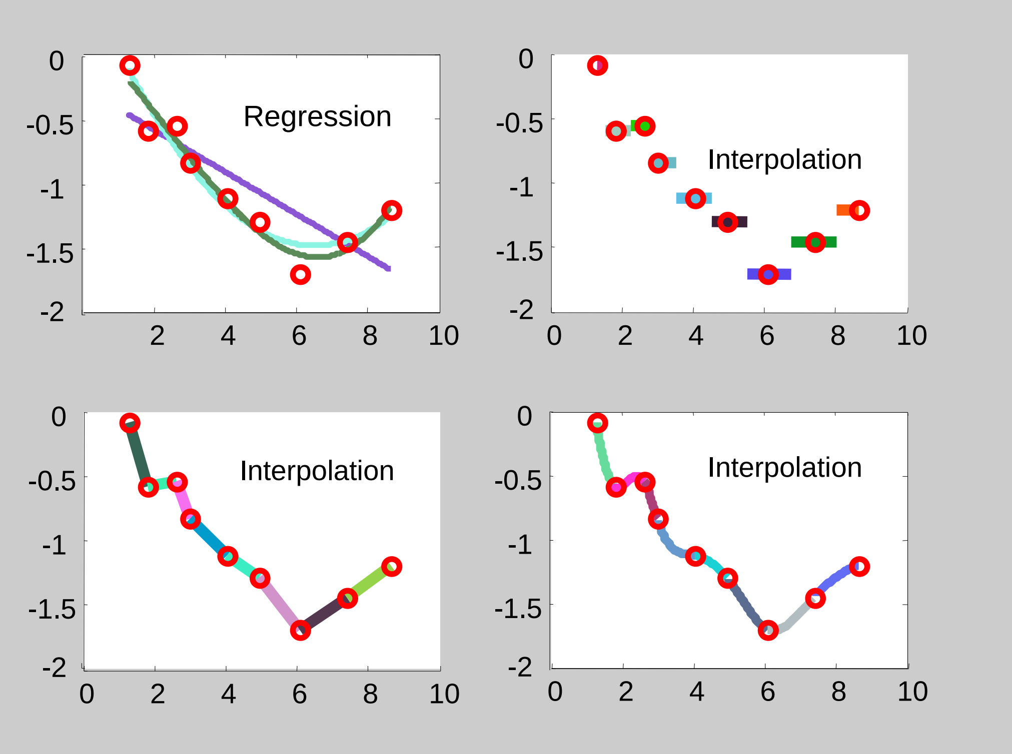
Let f^I denote such an interpolator.

The obtained function (f^R or f^I) is often used for *prediction*: given another value of x (with y unknown), *estimate/predict* "what is the corresponding value of y?"

Perspectives: Differences/similarities

Regression:

- Measured data is inaccurate in a random fashion.
- Data seems to exhibit "complex" variability, but that variability is not truly indicative of the underlying phenomena that produces the data
- Choose a "model" class of functions to represent the phenomena. It should be no more complex than necessary
- Get **approximate** fit from this simple class of functions.



Interpolation:

- Measured data is believed to be accurate.
- Any complex variability is thought to be truly indicative of the underlying phenomena that produces the data
- Functional representation should pass through all data points; no reason to expect wild variability between points.
- The smoothness of the interpolating function used depends on assumptions about the phenomena that produces the data.

Regression or Interpolation?

Accurate, fast approximation to

$$f(r) := \int_0^r e^{-x^{3/2}} \tan^{-1} x \; dx$$

Interpolation: it makes sense that the fitting function should pass through the training points

over the range $r \in [0, \quad 5]$. Data, $(r_i, f_i)_{i=1,\cdots,N}$ is obtained by careful numerical integration, is repeatable, and not subject to any appreciable error.

Regression: the "experimental" aspect hints at errors and some degree of non-repeatability in the training data due to uncontrolled/unknown factors.

Rebound velocity, as a function of impact velocity for a composite baseball bat and baseball, $r(v)$. Data, $(v_i, r_i)_{i=1,\cdots,N}$ is obtained from experimental lab apparatus.

Polynomial Interpolation

Linear interpolation

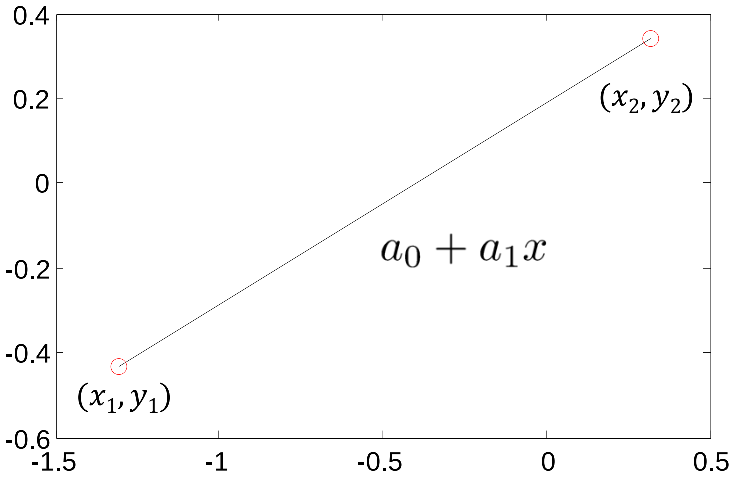
A linear (affine) polynomial has 2 parameters

$$p_a(x) = a_0 + a_1 x$$

Given 2 data points, $(x_1, y_1), (x_2, y_2)$ with the $x_1 \neq x_2$, there exists a unique choice of parameters (a_0, a_1) so that $p_a(xi) = y_i$ for $i = 1, 2$.

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$Ax = b$$



quadratic interpolation

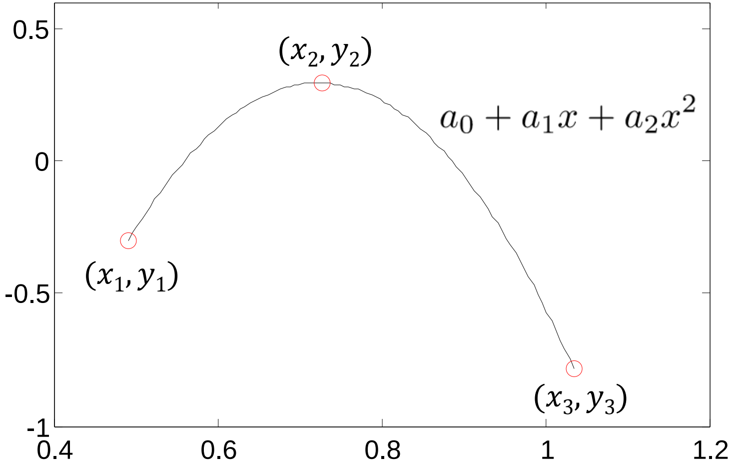
A quadratic polynomial has 3 parameters

$$p_a(x) = a_0 + a_1 x + a_2 x^2$$

Given 3 data points, $(x_1, y_1), (x_2, y_2), (x_3, y_3)$, with the $\{x_i\}_{i=1,2,3}$ all distinct from one another, there exists a unique choice of parameters (a_0, a_1, a_2) so that $p_a(xi) = y_i$ for $i = 1, 2, 3$.

$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$Ax = b$$



Cubic interpolation

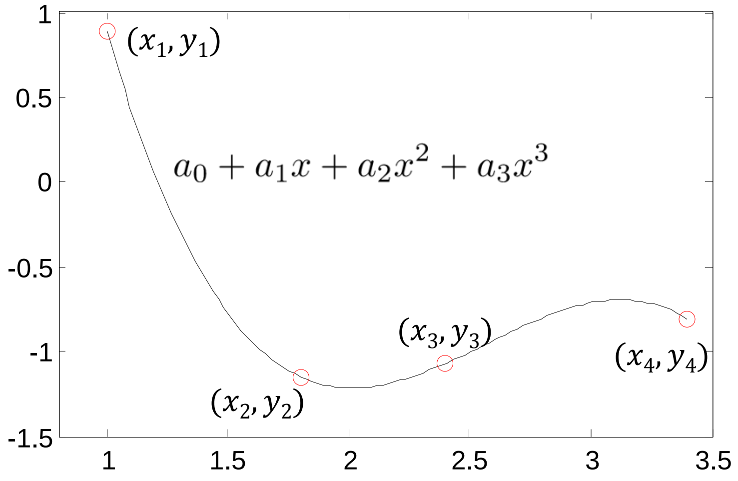
A cubic polynomial has 4 parameters

$$p_a(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

Given 4 data points, $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$, with the $\{x_i\}_{i=1,2,3,4}$ all distinct from one another, there exists a unique choice of parameters (a_0, a_1, a_2, a_3) so that $p_a(xi) = y_i$ for $i = 1, 2, 3, 4$.

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \\ 1 & x_4 & x_4^2 & x_4^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

$$Ax = b$$



Polynomial interpolation

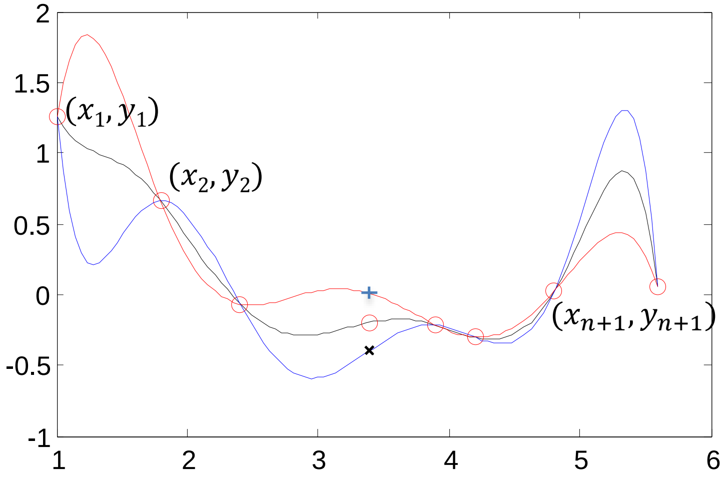
More generally, an n^{th} degree polynomial has $n+1$ parameters

$$p_a(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

Given $n+1$ data points, $(x_1, y_1), (x_2, y_2), \cdots, (x_{n+1}, y_{n+1})$, with the $\{x_i\}_{i=1,2,\cdots,(n+1)}$ all distinct from one another, there exists a unique choice of parameters $(a_0, a_1, a_2, \cdots, a_n)$ so that $p_a(xi) = y_i$ for $i = 1, 2, \cdots, n+1$.

$$\underbrace{\begin{bmatrix} 1 & x_1 & \cdots & x_1^n \\ 1 & x_2 & \cdots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n+1} & \cdots & x_{n+1}^n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n+1} \end{bmatrix}}_{b}$$

$$Ax = b$$



Behavior of Polynomial Interpolation

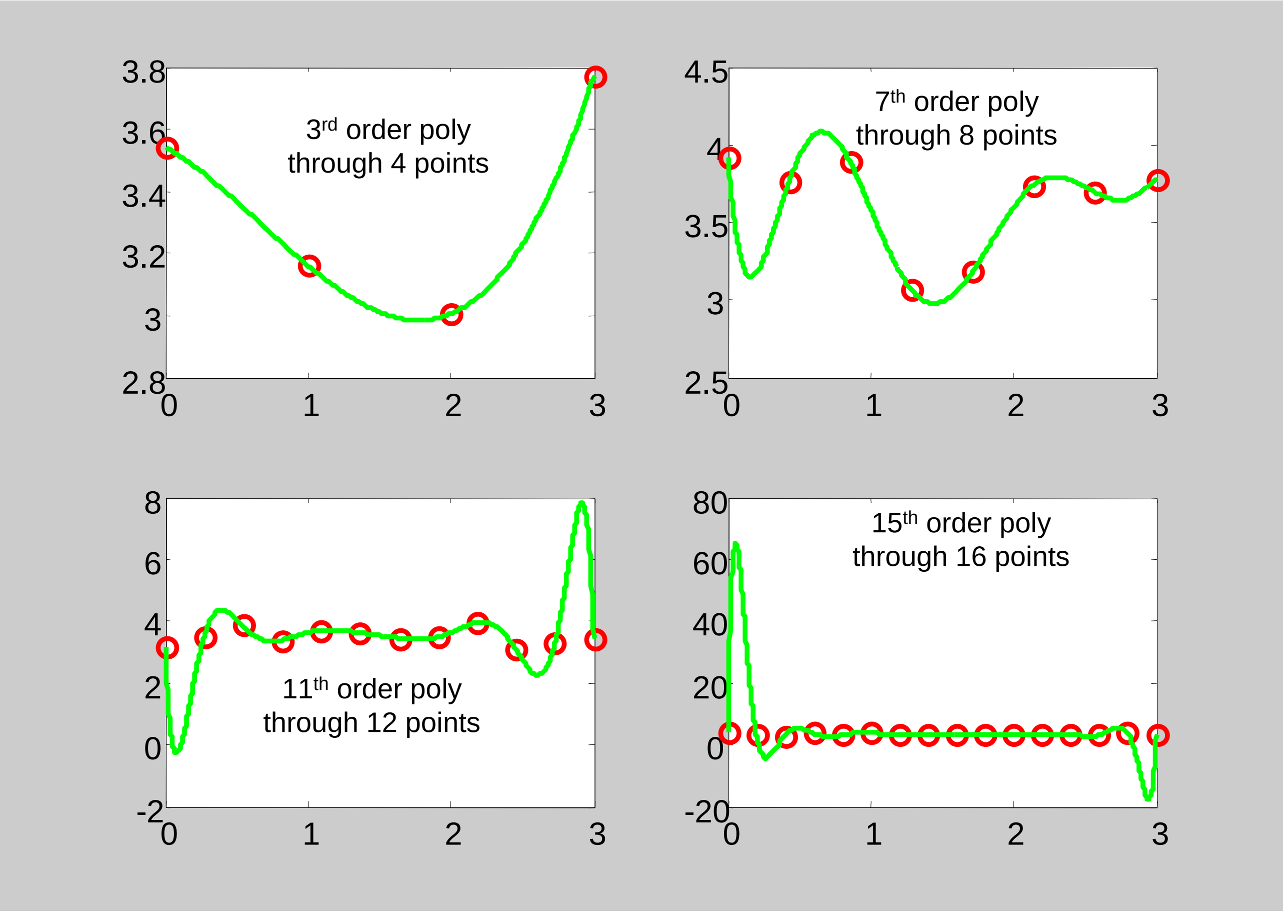
For large n, this polynomial tends to have large oscillations near the end points.

The behavior of the interpolating function, between points is unexpected and non-intuitive.

For evenly-spaced x, the prediction (ie., interpolating polynomial's value) at the midpoint between x_1 and x_2 becomes highly sensitive to $y_{i=n/2}$.

n	5	7	9	11	13	15
$Sens$	1.2	1.5	3.3	8.6	24	72

Polynomial interpolating functions using large n are generally thought of as *bad predictors*



Spline interpolation: linear and cubic

linear-spline

Given: data, $(x_i, y_i)_{i=1,\ldots,n}$. The (x_i) are distinct and sorted, so that $x_i < x_{i+1}$.

A *linear-spline interpolation* consists of $(n-1)$ linear (affine) functions, defined separately on each interval $[x_i \;\; x_{i+1}]$.

- 1st function connects (x_1, y_1) to (x_2, y_2) with a straight line,
- 2nd function connects (x_2, y_2) to (x_3, y_3) with a straight line,
- kth function connects (x_k, y_k) to (x_{k+1}, y_{k+1}) with a straight line
  - For X in $[x_k \;\; x_{k+1}]$ interpolation function (relating X to Y) is easily expressed as

$$Y = y_k + \frac{y_{k+1} - y_k}{x_{k+1} - x_k}(X - x_k)$$

- (n-1)th function connects (x_{n-1}, y_{n-1}) to (x_n, y_n) with a straight line.

Viewed as a whole: the interpolation function is piecewise-linear

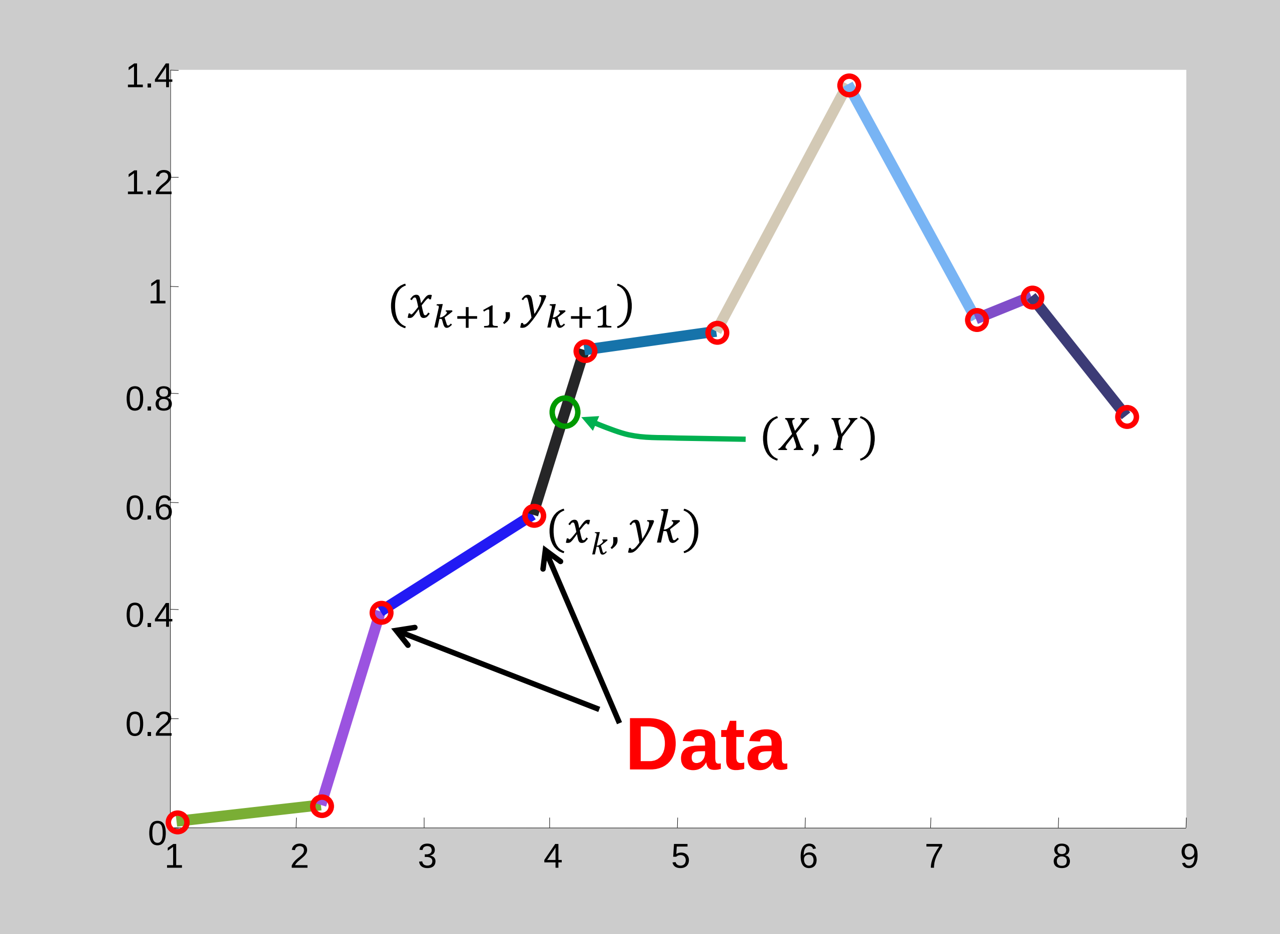
Linear-spline (or just "linear") interpolation

$$Y = y_k + \frac{y_{k+1} - y_k}{x_{k+1} - x_k}\left(X - x_k\right)$$

Starting height

Slope in this interval

"Run"



Cubic-spline

Given: data, $(x_i, y_i)_{i=1,\ldots,n}$. The (xi) are distinct and sorted, so that $x_i < x_{i+1}$.

A *cubic-spline interpolation* consists of cubic polynomial functions, defined on each interval $[xi \;\; x_{i+1}]$.

- 1st function connects (x_1, y_1) to (x_2, y_2) with cubic polynomial,
- 2nd function connects (x_2, y_2) to (x_3, y_3) with cubic polynomial,
- kth function connects (x_k, yk) to (x_{k+1}, y_{k+1}) with cubic polynomial,
- (n-1)th function connects (x_{n-1}, y_{n-1}) to (x_n, yn) with cubic polynomial

Constraints

@ x_1 $\quad f_1(x_1) = y_1$

@ x_n $\quad f_{n-1}(x_n) = y_n$

@ x_k $\quad f_{k-1}(x_k) = y_k \quad f'_{k-1}(x_k) = f'_k(x_k)$

$(2 \le k \le n-1)$ $\quad f_k(x_k) = y_k \quad f''_{k-1}(x_k) = f''_k(x_k)$

kth function is parametrized by 4 coefficients (to be determined by the constraints)

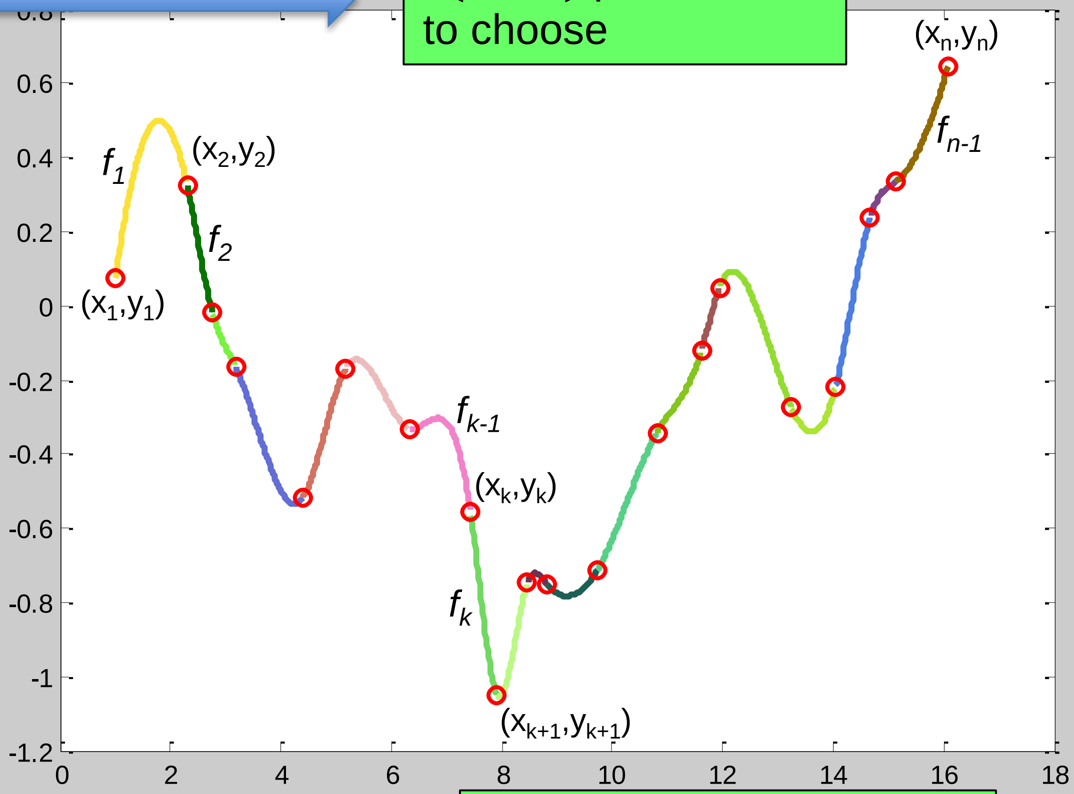
$$f_k(x) = a_0 + a_1(x - x_k) + a_2(x - x_k)^2 + a_3(x - x_k)^3$$

Cubic spline

$(n-1)$ cubic polynomial functions, each defined on $[xi\ x_{i+1}]$.

–k[th] function connects (x_k, yk) to (x_{k+1}, y_{k+1})

4$(n-1)$ parameters to choose



Constraints

@ x_1

1 equation

$$f_1(x_1) = y_1$$

@ x_k $(2 \le k \le n-1)$

$$f_{k-1}(x_k) = y_k$$
$$f_k(x_k) = y_k$$
$$f'_{k-1}(x_k) = f'_k(x_k)$$
$$f''_{k-1}(x_k) = f''_k(x_k)$$

$4(n-2)$ equations

@ x_n

1 equation

$$f_{n-1}(x_n) = y_n$$

$4(n-2) + 2$ equations

Two Extra conditions

As there are fewer equations that unknowns (coefficients of the cubic functions), the matching conditions (constraints) are not enough to uniquely determine the cubic functions.

Two more conditions can be imposed. Four common approaches are

– Natural

$$f_1''(x_1) = 0, \qquad f_{n-1}''(x_n) = 0$$

– Periodic

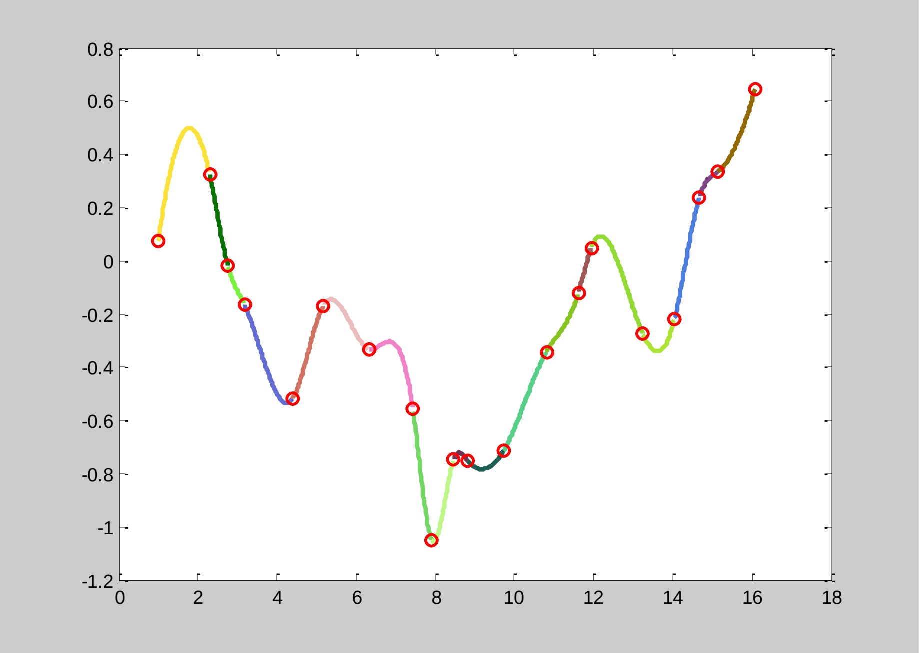
$$f_1'(x_1) = f_{n-1}'(x_n), \;\; f_1''(x_1) = f_{n-1}''(x_n)$$

– Not-a-knot $\quad f_1'''(x_2) = f_2'''(x_2),$

$$f_{n-2}'''(x_{n-1}) = f_{n-1}'''(x_{n-1})$$

– Specified end-slope, given values v_1, v_n

$$f_1'(x_1) = v_1, \quad f_{n-1}'(x_n) = v_n$$



Setting up the equations

Values and derivatives of cubic

What are the values and derivative of a cubic

$$f(x) = a_0 + a_1(x - \bar{x}) + a_2(x - \bar{x})^2 + a_3(x - \bar{x})^3$$

$$f'(x) = a_1 + 2a_2(x - \bar{x}) + 3a_3(x - \bar{x})^2$$

$$f''(x) = 2a_2 + 6a_3(x - \bar{x})$$

$$f'''(x) = 6a_3$$

$$\begin{bmatrix} f(x) \\ f'(x) \\ f''(x) \\ f'''(x) \end{bmatrix} = \begin{bmatrix} 1 & x - \bar{x} & (x - \bar{x})^2 & (x - \bar{x})^3 \\ 0 & 1 & 2(x - \bar{x}) & 3(x - \bar{x})^2 \\ 0 & 0 & 2 & 6(x - \bar{x}) \\ 0 & 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Values and derivatives of f_k

For f_k, the value of $\bar{x} = x_k$, so

$$\begin{aligned} f_k(x) &= a_0 + a_1(x - x_k) + a_2(x - x_k)^2 + a_3(x - x_k)^3 \\ f_k'(x) &= a_1 + 2a_2(x - x_k) + 3a_3(x - x_k)^2 \\ f_k''(x) &= 2a_2 + 6a_3(x - x_k) \\ f_k'''(x) &= 6a_3 \end{aligned}$$

$$\begin{bmatrix} f_k(x_k) \\ f_k'(x_k) \\ f_k''(x_k) \\ f_k'''(x_k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}_k \qquad \begin{bmatrix} f_k(x_{k+1}) \\ f_k'(x_{k+1}) \\ f_k''(x_{k+1}) \\ f_k'''(x_{k+1}) \end{bmatrix} = \begin{bmatrix} 1 & h_k & h_k^2 & h_k^3 \\ 0 & 1 & 2h_k & 3h_k^2 \\ 0 & 0 & 2 & 6h_k \\ 0 & 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}_k$$

$@x = x_k$ $\qquad$ $@x = x_{k+1}$ $\qquad$ $h_k := x_{k+1} - x_k$

Constraints

$$@x_1, f_1(x_1) = y_1$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}_1 = y_1$$

$$\boxed{(a_0)_1 = y_1}$$

$$@x_n, f_{n-1}(x_n) = y_n$$

$$\begin{bmatrix} 1 & h_{n-1} & h_{n-1}^2 & h_{n-1}^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}_{n-1} = y_n$$

$$@x_k, 2 \leq k \leq (n-1)$$

$$f_{k-1}(x_k) = y_k$$

$$f_k(x_k) = y_k$$

$$f'_{k-1}(x_k) = f'_k(x_k)$$

$$f''_{k-1}(x_k) = f''_k(x_k)$$

$$\begin{bmatrix} 1 & h_{k-1} & h_{k-1}^2 & h_{k-1}^3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2h_{k-1} & 3h_{k-1}^2 & 0 & -1 & 0 & 0 \\ 0 & 0 & 2 & 6h_{k-1} & 0 & 0 & -2 & 0 \end{bmatrix} \begin{bmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}_{k-1} \\ \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}_k \end{bmatrix} = \begin{bmatrix} y_k \\ y_k \\ 0 \\ 0 \end{bmatrix}$$

$$\boxed{(a_0)_k = y_k}$$

Rewrite Constraints

$$(a_0)_1 = y_1$$
$$(a_0)_k = y_k, 2 \le k \le (n-1)$$

$$\boxed{(a_0)_k = y_k, \quad \forall k \quad (1 \le k \le (n-1))}$$

$$@x_n, f_{n-1}(x_n) = y_n$$

$$\begin{bmatrix} h_{n-1} & h_{n-1}^2 & h_{n-1}^3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}_{n-1} = y_n - y_{n-1}$$

$$@x_k, 2 \le k \le (n-1)$$

$$f_{k-1}(x_k) = y_k$$
$$f'_{k-1}(x_k) = f'_k(x_k)$$
$$f''_{k-1}(x_k) = f''_k(x_k)$$

$$\begin{bmatrix} h_{k-1} & h_{k-1}^2 & h_{k-1}^3 & 0 & 0 & 0 \\ 1 & 2h_{k-1} & 3h_{k-1}^2 & -1 & 0 & 0 \\ 0 & 2 & 6h_{k-1} & 0 & -2 & 0 \end{bmatrix} \begin{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_{k-1} \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_k \end{bmatrix} = \begin{bmatrix} y_k - y_{k-1} \\ 0 \\ 0 \end{bmatrix}$$

Assembling the equations ($n = 6$)

$$
\left[\begin{array}{cccc}
\begin{bmatrix} h_1 & h_1^2 & h_1^3 & 0 & 0 & 0 \\ 1 & 2h_1 & 3h_1^2 & -1 & 0 & 0 \\ 0 & 2 & 6h_1 & 0 & -2 & 0 \end{bmatrix} & & & \\
& \begin{bmatrix} h_2 & h_2^2 & h_2^3 & 0 & 0 & 0 \\ 1 & 2h_2 & 3h_2^2 & -1 & 0 & 0 \\ 0 & 2 & 6h_2 & 0 & -2 & 0 \end{bmatrix} & & \\
& & \begin{bmatrix} h_3 & h_3^2 & h_3^3 & 0 & 0 & 0 \\ 1 & 2h_3 & 3h_3^2 & -1 & 0 & 0 \\ 0 & 2 & 6h_3 & 0 & -2 & 0 \end{bmatrix} & \\
& & & \begin{bmatrix} h_4 & h_4^2 & h_4^3 & 0 & 0 & 0 \\ 1 & 2h_4 & 3h_4^2 & -1 & 0 & 0 \\ 0 & 2 & 6h_4 & 0 & -2 & 0 \end{bmatrix} \\
& & & \begin{bmatrix} h_5 & h_5^2 & h_5^3 \end{bmatrix}
\end{array}\right]
\begin{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_1 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_2 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_3 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_4 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_5 \end{bmatrix}
=
\begin{matrix} \begin{bmatrix} y_2 - y_1 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} y_3 - y_2 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} y_4 - y_3 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} y_5 - y_4 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} y_6 - y_5 \end{bmatrix} \end{matrix}
$$

Two more conditions

Assembling the equations (arbitrary n)

$$\begin{bmatrix} h_1 & h_1^2 & h_1^3 & 0 & 0 & 0 \\ 1 & 2h_1 & 3h_1^2 & -1 & 0 & 0 \\ 0 & 2 & 6h_1 & 0 & -2 & 0 \end{bmatrix}$$

```
Cols=
3*(k-1)+1:3k+3
```

```
Rows=
3*(k-1)+1:3k
```

$$\begin{bmatrix} h_k & h_k^2 & h_k^3 & 0 & 0 & 0 \\ 1 & 2h_k & 3h_k^2 & -1 & 0 & 0 \\ 0 & 2 & 6h_k & 0 & -2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} h_{n-2} & h_{n-2}^2 & h_{n-2}^3 & 0 & 0 & 0 \\ 1 & 2h_{n-2} & 3h_{n-2}^2 & -1 & 0 & 0 \\ 0 & 2 & 6h_{n-2} & 0 & -2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} h_{n-1} & h_{n-1}^2 & h_{n-1}^3 \end{bmatrix}$$

Two more conditions

$$\begin{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_1 \\ \vdots \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_{n-1} \end{bmatrix}$$

Incorporating extra conditions

Two Extra conditions

As there are fewer equations that unknowns (coefficients of the cubic functions), the matching conditions are not enough to uniquely determine the cubic functions.

Two more conditions can be imposed. Four common approaches are

- Natural

$$f_1''(x_1) = 0, \qquad f_{n-1}''(x_n) = 0$$

- Periodic

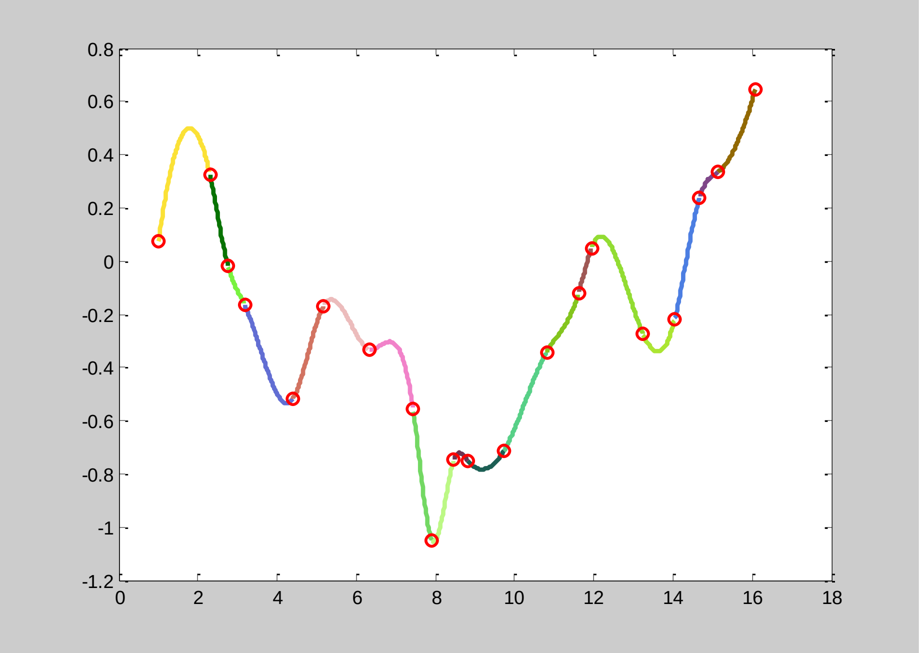
$$f_1'(x_1) = f_{n-1}'(x_n), \;\; f_1''(x_1) = f_{n-1}''(x_n)$$

- Not-a-knot

$$f_1'''(x_2) = f_2'''(x_2),$$
$$f_{n-2}'''(x_{n-1}) = f_{n-1}'''(x_{n-1})$$

- Specified end-slope, given values v_1, v_n

$$f_1'(x_1) = v_1, \quad f_{n-1}'(x_n) = v_n$$



Values and derivatives of f_k

For f_k, the value of $\bar{x} = x_k$, so

$$
\begin{aligned}
f_k(x) &= a_0 + a_1(x - x_k) + a_2(x - x_k)^2 + a_3(x - x_k)^3 \\
f_k'(x) &= a_1 + 2a_2(x - x_k) + 3a_3(x - x_k)^2 \\
f_k''(x) &= 2a_2 + 6a_3(x - x_k) \\
f_k'''(x) &= 6a_3
\end{aligned}
$$

$$
\begin{bmatrix} f_k(x_k) \\ f_k'(x_k) \\ f_k''(x_k) \\ f_k'''(x_k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}_k
\qquad
\begin{bmatrix} f_k(x_{k+1}) \\ f_k'(x_{k+1}) \\ f_k''(x_{k+1}) \\ f_k'''(x_{k+1}) \end{bmatrix} = \begin{bmatrix} 1 & h_k & h_k^2 & h_k^3 \\ 0 & 1 & 2h_k & 3h_k^2 \\ 0 & 0 & 2 & 6h_k \\ 0 & 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}_k
$$

$@x = x_k$ $\qquad$ $@x = x_{k+1}$ $\qquad$ $h_k := x_{k+1} - x_k$

Extra equations for Natural spline

Conditions for Natural Spline

$$f_1''(x_1) = 0 \qquad f_{n-1}''(x_n) = 0$$

Expressions

$$f_1''(x_1) = 2(a_2)_1$$

$$f_{n-1}''(x_n) = 2(a_2)_{n-1} + 6h_{n-1}(a_3)_{n-1}$$

$$\begin{bmatrix} \begin{bmatrix} 0 & 2 & 0 \end{bmatrix} & & & & \\ & & & & \begin{bmatrix} 0 & 2 & 6h_{n-1} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_1 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_2 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_3 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_4 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_5 \end{bmatrix} = \begin{matrix} 0 \\ 0 \end{matrix}$$

Extra equations for Periodic spline

Conditions for Periodic Spline

$$f_1'(x_1) = f_{n-1}'(x_n) \qquad f_1''(x_1) = f_{n-1}''(x_n)$$

Expressions

$$\begin{aligned} f_1'(x_1) &= (a_1)_1 \\ f_{n-1}'(x_n) &= (a_1)_{n-1} + 2h_{n-1}(a_2)_{n-1} + 3h_{n-1}^2(a_3)_{n-1} \end{aligned}$$

$$\begin{aligned} f_1''(x_1) &= 2(a_2)_1 \\ f_{n-1}''(x_n) &= 2(a_2)_{n-1} + 6h_{n-1}(a_3)_{n-1} \end{aligned}$$

$$\left[\begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \end{bmatrix} \quad \cdots \quad \begin{bmatrix} 1 & 2h_{n-1} & 3h_{n-1}^2 \\ 0 & 2 & 6h_{n-1} \end{bmatrix}\right] \begin{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_1 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_2 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_3 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_4 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Extra equations for Not-a-knot

Conditions for Not-a-knot spline

$$f_1'''(x_2) = f_2'''(x_2) \qquad f_{n-2}'''(x_{n-1}) = f_{n-1}'''(x_{n-1})$$

Expressions

$$f_1'''(x_2) = 6(a_3)_1$$

$$f_2'''(x_2) = 6(a_3)_2$$

$$f_{n-2}'''(x_{n-1}) = 6(a_3)_{n-2}$$

$$f_{n-1}'''(x_{n-1}) = 6(a_3)_{n-1}$$

$$\begin{bmatrix} \begin{bmatrix} 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} 0 & 0 & -6 \end{bmatrix} & \\ & \begin{bmatrix} 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} 0 & 0 & -6 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_1 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_2 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_3 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_4 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_5 \end{bmatrix} = \begin{matrix} 0 \\ 0 \end{matrix}$$

Extra equations for Specified end-slope

Conditions for specified end-slope

$$f_1'(x_1) = v_1, \quad f_{n-1}'(x_n) = v_n$$

Expressions

$$f_1'(x_1) = (a_1)_1$$

$$f_{n-1}'(x_n) = (a_1)_{n-1} + 2h_{n-1}(a_2)_{n-1} + 3h_{n-1}^2(a_3)_{n-1}$$

$$\begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} & \\ & \begin{bmatrix} 1 & 2h_{n-1} & 3h_{n-1}^2 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_1 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_2 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_3 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_4 \\ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}_5 \end{bmatrix} = \begin{matrix} v_1 \\ v_n \end{matrix}$$

Implementation

Need one function to transform $(x_i, y_i)_{i=1,\ldots,n}$ into the coefficients $\left(\{a_0, a_1, a_2, a_3\}_{i=1,\ldots,(n-1)}\right)$. Just build the arrays, and use **** (backslash)

Need another function to evaluate the "spline" at an arbitrary value of x. This function needs

- the coefficients, $\left(\{a_0, a_1, a_2, a_3\}_{i=1,\ldots,(n-1)}\right)$,
- the $(x_i)_{i=1,\ldots,n}$ samples, and
- the x-values for which the evaluation should take place.

The commands

`spline, ppval`

implement the ideas put forth here, in a more efficient manner. Only the "not-a-knot" and "specified end-slope" conditions are available.