# CS537

# Numerical Analysis

## Lecture 6

## Approximation by Spline Functions

Professor Jun Zhang

Department of Computer Science
University of Kentucky
Lexington, KY 40206-0046

October 30, 2008

# Lower Degree Splines

Given a set of data, it is possible to construct a polynomial to interpolate these data. The disadvantage of higher order polynomials is its oscillation behavior

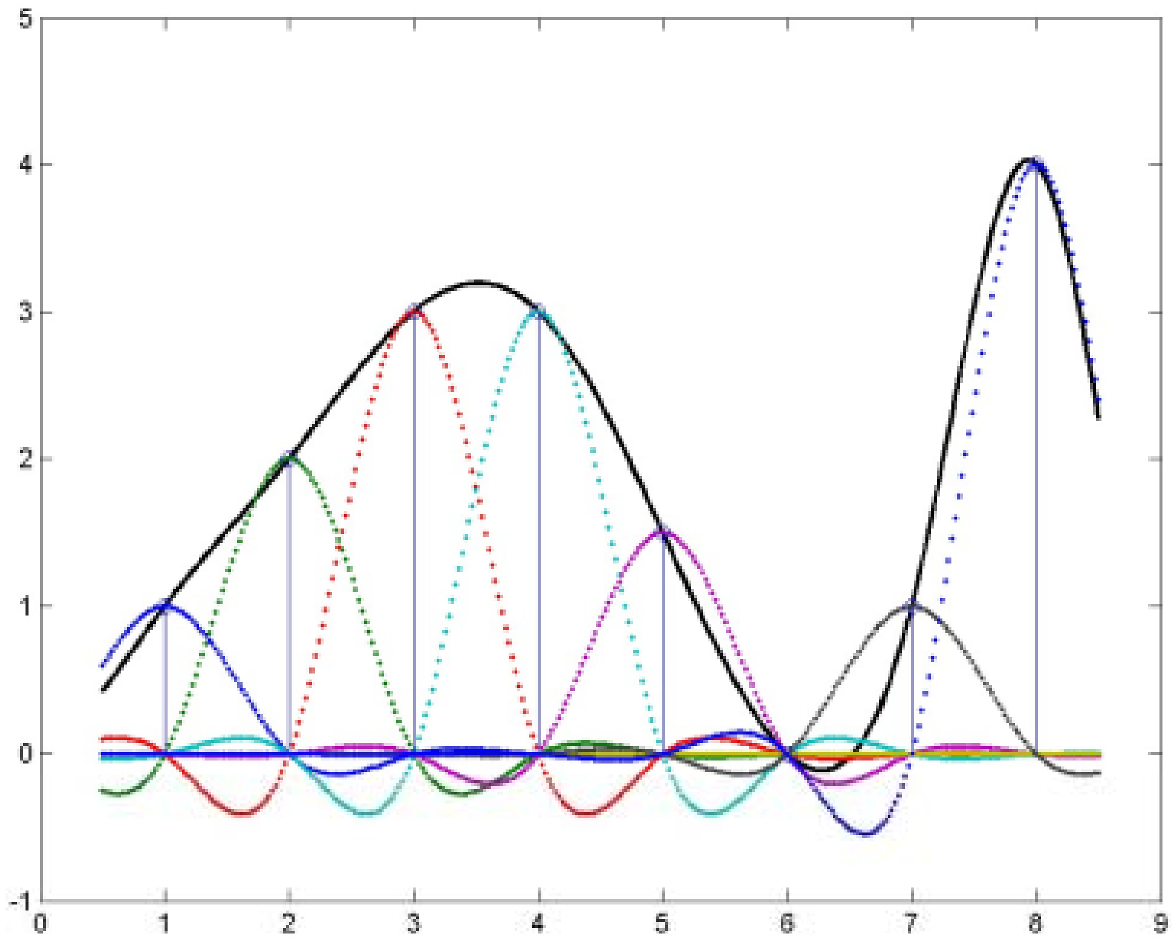
A spline function is a function that consists of polynomial pieces joined together with certain smoothness conditions

First degree spline function is a polygonal function with linear polynomials joined together to achieve continuity

The points $t_0, t_1, \ldots, t_n$ at which the function changes its character are called knots

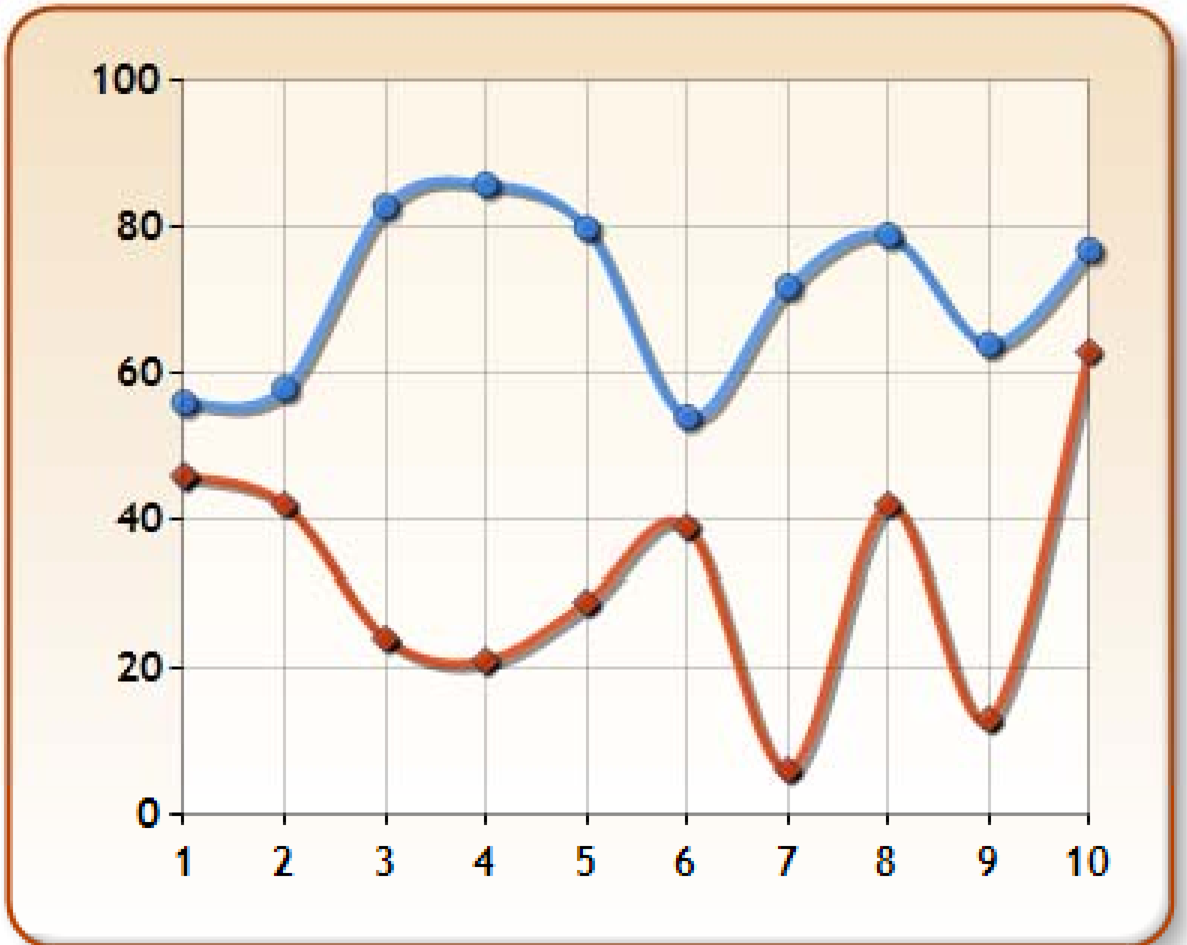
Note that **knots** do not have to be nodes

# High Order Polynomials




Interpolation uses just one polynomial which may be oscillatory if the order of the polynomial is high

# Splines




A spline function is a function that consists of polynomial pieces joined together with certain smoothness conditions

# Piecewise Linear Polynomial

A piecewise linear function for the spline of degree 1 can be written as

$$S(x) = \begin{cases} S_0(x) & x \in [t_0, t_1] \\ S_1(x) & x \in [t_1, t_2] \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [t_{n-1}, t_n] \end{cases}$$

where

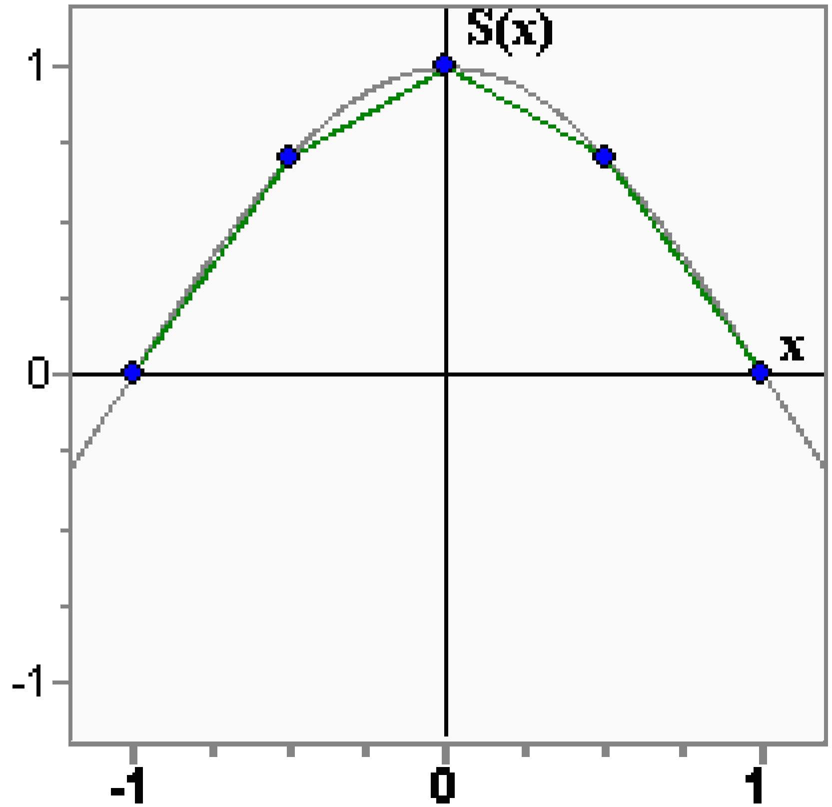
$$S_i(x) = a_i x + b_i$$

Is a linear polynomial

The knots $t_0, t_1, \ldots, t_n$ and the coefficients $a_i, b_i, i = 0, 1, \ldots, n-1$ have to be known in order to evaluate $S(x)$

First to determine which interval $x$ lies, then evaluate the linear function defined on that interval

# Piecewise Linear Polynomial




To evaluate $S(x)$**,** first to determine which interval $x$ lies, then evaluate the linear function defined on that interval

# Spline of Degree 1

A function $S(x)$ is called a spline of degree 1 if

1. The domain of $S$ is an interval $[a,b]$

2. $S$ is continuous on $[a,b]$

3. There is a partitioning of the interval $a = t_0 < t_1 < \ldots < t_n = b$ such that $S$ is a linear polynomial on each subinterval $[t_i,t_{i+1}]$

   For outside part of the interval $[a,b]$, we define $S(x) = S_0(x)$ when $x < a$, and $S(x) = S_{n-1}(x)$ when $x > b$

   It is important that the spline of degree 1 be continuous at the knots, i.e., the left limit and the right limit are equal

$$\lim_{x \to t_i^+} f(x) = \lim_{x \to t_i^-} f(x) = f(t_i)$$

## An Example

Determine if the following function is a first degree spline

$$S(x) = \begin{cases} x & -1 \le x \le 0.5 \\ 0.5x + 2(x - 0.5) & 0.5 \le x \le 2 \\ x + 1.5 & 2 \le x \le 4 \end{cases}$$

Each linear function is continuous on the subinterval it is defined. We need to verify if they are continuous at the two interior knots $x = 0.5$ and $x = 2$

$$\lim_{x \to 0.5^-} S(x) = \lim_{x \to 0.5^-} x = 0.5$$

$$\lim_{x \to 0.5^+} S(x) = \lim_{x \to 0.5^+} 0.5x + 2(x - 0.5) = 0.25$$

The function is not a spline of degree 1, as

$$\lim_{x \to 0.5^-} S(x) \ne \lim_{x \to 0.5^+} S(x)$$

i.e., $S(x)$ is discontinuous at the knot $x = 0.5$

# Construct Spline of Degree 1

Given a data set with $t_0 < t_1 < \ldots < t_n$

| $x$ | $t_0$ | $t_1$ | $\cdots$ | $t_n$ |
|---|---|---|---|---|
| $y$ | $y_0$ | $y_1$ | $\cdots$ | $y_n$ |

A linear polynomial can be constructed using two pairs of neighboring data

First compute the slope of the line as

$$m = \frac{y_{i+1} - y_i}{t_{i+1} - t_i}$$

The straight line equation is given by the point-slope formula as

$$S_i(x) = y_i + m_i(x - t_i)$$

It is easy to see that we have $2n$ degrees of freedom $a_i$ and $b_i$ and $2n$ conditions. So the construction of first degree spline is guaranteed

# Modulus of Continuity

The modulus of continuity of a function $f$ is defined as, for $a \le u \le v \le b$

$$\omega(f;h) = \sup\left\{ \left| f(u) - f(v) \right| \quad : \quad \left| u - v \right| \le h \right\}$$

The quantity is the largest variation of $f$ over a small interval of size $h$. It measures how much $f$ can change in such an interval

If $f$ is continuous on $[a,b]$ then

$$\lim_{h \to 0} \omega(f;h) = 0$$

If $f$ is differentiable on $[a,b]$ then

$$\left| f(u) - f(v) \right| = \left| f'(c)(u-v) \right| \le M_1 \left| u - v \right| \le M_1 h$$

where $M_1$ is the maximum value of $|f'(x)|$ on $(a,b)$. It follows that

$$\omega(f;h) \le M_1 h$$

# Theorems

If $p$ is the first degree polynomial that interpolates a function $f$ at the end points of an interval $[a,b]$, then with $h = b - a$, we have

$$|f(x) - p(x)| \le \omega(f;h) \qquad (a \le x \le b)$$

Note that the linear function passes through $[a,b]$ can be written as

$$p(x) = \left(\frac{x-a}{b-a}\right) f(b) + \left(\frac{b-x}{b-a}\right) f(a)$$

Hence

$$f(x) - p(x) = \left(\frac{x-a}{b-a}\right)\left[f(x) - f(b)\right] + \left(\frac{b-x}{b-a}\right)\left[f(x) - f(a)\right]$$

Note that

$$\left(\frac{x-a}{b-a}\right) f(x) + \left(\frac{b-x}{b-a}\right) f(x) = f(x)$$

The result follows immediately

## Accuracy of First Degree Spline

Let $p$ be a first degree spline having knots $a = x_0 < x_1 < \ldots < x_n = b$. If $p$ interpolates a function $f$ at these knots, then with $h = \max_i(x_i - x_{i-1})$ we have

$$|f(x) - p(x)| \le \omega(f; h) \qquad (a \le x \le b)$$

If more knots are inserted such that the maximum spacing $h$ goes to zero, the corresponding first degree spline will converge uniformly to $f$

If $f'$ or $f''$ exist and are continuous, we have

$$|f(x) - p(x)| \le M_1 \frac{h}{2} \qquad (a \le x \le b)$$

$$|f(x) - p(x)| \le M_2 \frac{h^2}{8} \qquad (a \le x \le b)$$

where $M_1$ and $M_2$ are the maximum values of $f'$ and $f''$ on $(a,b)$, respectively

## Accuracy of First Degree Spline

From the estimates of the accuracy of the first-degree spline interpolation

$$|f(x) - p(x)| \le M_1 \frac{h}{2} \qquad (a \le x \le b)$$

$$|f(x) - p(x)| \le M_2 \frac{h^2}{8} \qquad (a \le x \le b)$$

It is obvious that when *h* approaches zero, with more knot points inserted, the error of the first-degree spline goes to zero

It follows that the spline interpolation can be as accurate as needed, by inserting sufficient number of knot points

The same cannot be said about the polynomial interpolation, because of the oscillations associated with high order polynomials

## Second Degree Spline

A function $Q$ is a second degree spline if

1.) the domain of $Q$ is an interval $[a,b]$

2.) $Q$ and $Q'$ are continuous on $[a,b]$

3.) there are points $t_i$ such that $a = t_0 < t_1 < \ldots < t_n = b$ and $Q$ is a polynomial of degree at most $2$ on each subinterval $[t_i, t_{i-1}]$

A quadratic spline is a continuously differentiable piecewise quadratic function. It is a linear combination of basic functions $1, x, x^2$. The smoothness condition is stronger than that for the first degree spline

To determine a quadratic spline, we must verify the continuity of both $Q$ and $Q'$ at the know points

# Interpolating Quadratic Spline

Given a set of data

| $x$ | $t_0$ | $t_1$ | $t_2$ | $\cdots$ | $t_n$ |
|---|---|---|---|---|---|
| $y$ | $y_0$ | $y_1$ | $y_2$ | $\cdots$ | $y_n$ |

A quadratic spline $Q(x)$ can be constructed to interpolate these data, using $t_0, t_1, \ldots, t_n$ as the knots

A quadratic spline consisting of $n$ separate pieces of quadratic functions of the form

$$Q_i(x) = a_i x^2 + b_i x + c_i \qquad x \in [t_i, t_{i+1}]$$

There are $3n$ coefficients to be determined

The interpolating conditions $Q_i(t_i) = y_i$ and $Q_{i+1}(t_{i+1}) = y_{i+1}$ given $2n$ conditions. The continuity of $Q'(x)$ gives another $n - 1$ conditions at the interior knots. The last condition can be specified (arbitrarily) as $Q'(t_0) = 0$

## Computational Procedure

Construct a piecewise quadratic function

$$Q(x) = \begin{cases} Q_0(x) & x \in [t_0, t_1] \\ Q_1(x) & x \in [t_1, t_2] \\ \quad\vdots & \quad\vdots \\ Q_{n-1}(x) & x \in [t_{n-1}, t_n] \end{cases}$$

which is continuously differentiable on $[t_0, t_n]$ and $Q(t_i) = y_i$ for $0 \le i \le n$

The computational procedure is based on a recursive procedure on $Q'_i(t_i)$. Assuming $Q'(t_i) = z_i$, then we have

$$Q_i(x) = \frac{z_{i+1} - z_i}{2(t_{i+1} - t_i)}(x - t_i)^2 + z_i(x - t_i) + y_i$$

This definition can be verified by showing that $Q_i(t_i) = y_i$, $Q'_i(t_i) = z_i$, and $Q'_i(t_{i+1}) = z_{i+1}$. Note that

$$Q_i'(x) = \frac{z_{i+1} - z_i}{t_{i+1} - t_i}(x - t_i) + z_i$$

## Computational Procedure

The defined quadratic function $Q_i(x)$ does not necessarily satisfy the condition $Q_i(t_{i+1}) = y_{i+1}$ for $i = 0, 1,\ldots, n-1$. If we apply this condition, we have

$$y_{i+1} = \frac{z_{i+1} - z_i}{2(t_{i+1} - t_i)}(t_{i+1} - t_i)^2 + z_i(t_{i+1} - t_i) + y_i$$

After simplification, we obtain

$$z_{i+1} = -z_i + 2\left(\frac{y_{i+1} - y_i}{t_{i+1} - t_i}\right) \qquad (0 \le i \le n-1)$$

From this equation, we can compute the values of $z_0, z_1,\ldots, z_n$ recursively, by assigning an arbitrary value to $z_0$. Then the second degree spline is written as

$$Q_i(x) = \frac{z_{i+1} - z_i}{2(t_{i+1} - t_i)}(x - t_i)^2 + z_i(x - t_i) + y_i$$

For each interval $[t_i, t_{i+1}]$, $i = 0, 1,\ldots, n-1$.

# Subbotin Quadratic Spline

Knots are points where the spline function can change its form, nodes are points where the values of the function is specified

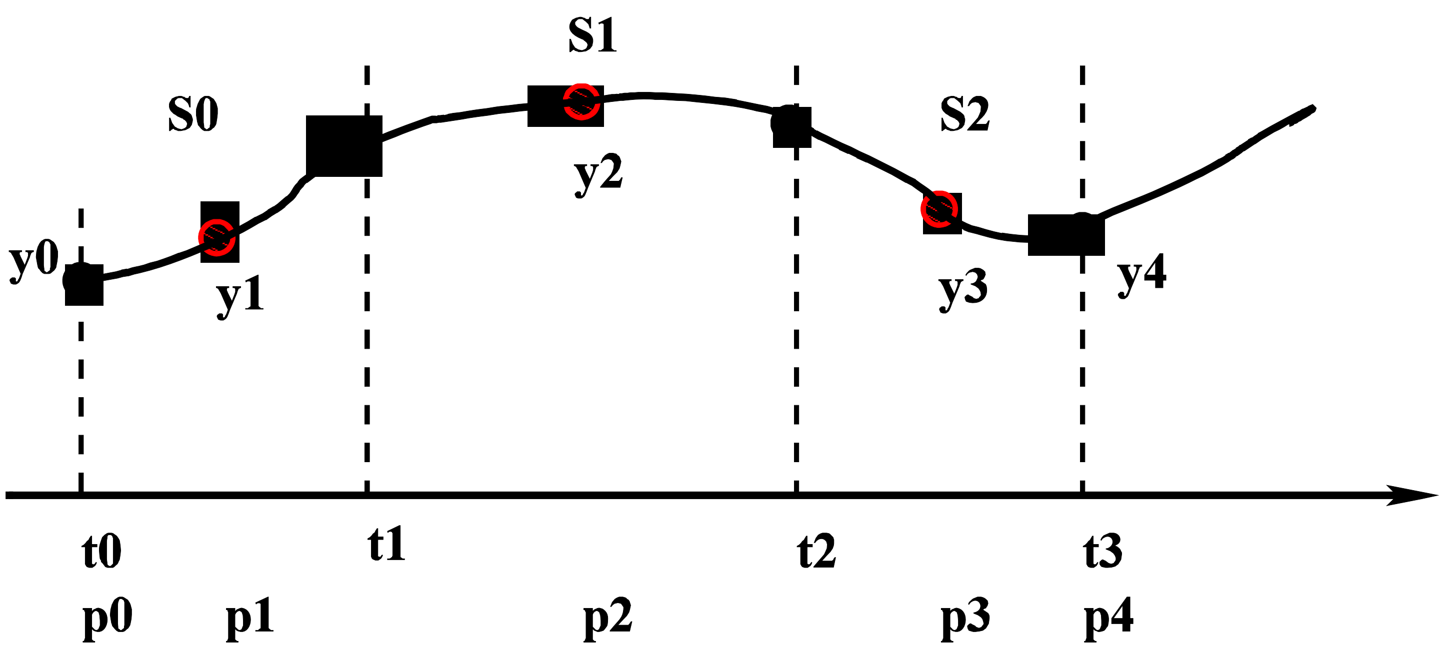
It is not necessary that the knots are nodes.

Subbotin suggests to use the midpoints of the knots as the (interpolation) nodes and the two end points. This yields $n + 2$ conditions. The continuity of $Q$ and $Q'$ give another $2(n - 1)$ conditions, for a total of $3n$ conditions. No free variable is required

Let the knots be $a = t_0 < t_1 < \ldots < t_n = b$, we define the interpolation nodes as

$$\begin{cases} p_0 = t_0 \qquad p_n = t_n \\ p_i = \frac{1}{2}(t_i + t_{i+1}) \qquad (1 \le i \le n) \end{cases}$$

# Subbotin Quadratic Spline




The knots be $\boldsymbol{a} = \boldsymbol{t}_0 < \boldsymbol{t}_1 < \ldots < \boldsymbol{t}_n = \boldsymbol{b}$, the interpolation nodes as

$$\begin{cases} p_0 = t_0 \qquad p_n = t_n \\ p_i = \frac{1}{2}(t_i + t_{i+1}) \qquad (1 \le i \le n) \end{cases}$$

# Subbotin Quadratic Spline (II)

The quadratic spline function $Q(x)$ should satisfy the interpolation conditions

$$Q(p_i) = y_i \qquad (0 \le i \le n+1)$$

Let $Q_i'(t_i) = z_i$, where $Q_i(x)$ is defined on the interval $[t_i, t_{i+1}]$, which can be written as

$$\begin{aligned} Q_i(x) &= y_{i+1} + \tfrac{1}{2}(z_{i+1} + z_i)(x - p_{i+1}) \\ &\quad + \tfrac{1}{2h_i}(z_{i+1} - z_i)(x - p_{i+1})^2 \end{aligned}$$

Where $h_i = t_{i+1} - t_i$. We can verify that

$$Q_i(p_{i+1}) = y_{i+1}, \quad Q_i'(t_i) = z_i, \quad Q_i'(t_{i+1}) = z_{i+1}$$

We also want to impose the continuity condition at the knots

$$\lim_{x \to t_i^-} Q_{i-1}'(x) = \lim_{x \to t_i^+} Q_i'(x) \quad (1 \le i \le n-1)$$

# Subbotin Quadratic Spline (III)

These conditions lead to a recursive formula

$$h_{i-1}z_{i-1} + 3(h_{i-1} + h_i)z_i + h_i z_{i+1} = 8(y_{i+1} - y_i)$$

for $i = 1,2,\ldots,n-1$

We impose the first and last interpolation conditions

$$Q(p_0) = y_0 \qquad\qquad Q(p_{n+1}) = y_{n+1}$$

Which yield two more equations

$$3h_0 z_0 + h_0 z_1 = 8(y_1 - y_0)$$

$$h_{n-1}z_{n-1} + 3h_{n-1}z_n = 8(y_{n+1} - y_n)$$

Now we have $n+1$ equations for the $n+1$ unknowns $z_0, z_1, \ldots, z_n$

We need to solve a linear system of the form $Ax = b$ with $A$ being a tridiagonal matrix

# An Example

Determine if the function is a quadratic spline

$$Q(x) = \begin{cases} -x^2 & x \le 0 \\ x & x > 0 \end{cases}$$

The interval can be viewed as $(-\infty,\infty)$ which is not a closed interval

Also, the derivative of $Q(x)$ is

$$Q'(x) = \begin{cases} -2x & x \le 0 \\ 1 & x > 0 \end{cases}$$

Hence, $Q'(x)$ is discontinuous at $x = 0$. It follows that $Q(x)$ is not a quadratic spline

How about the function

$$Q(x) = \begin{cases} 0.1x^2 & 0 \le x \le 1 \\ 9.3x^2 - 18.4x + 9.2 & 1 \le x \le 1.3 \end{cases}$$

## Natural Cubic Spline

Very high order polynomials oscillate and give undesired feature

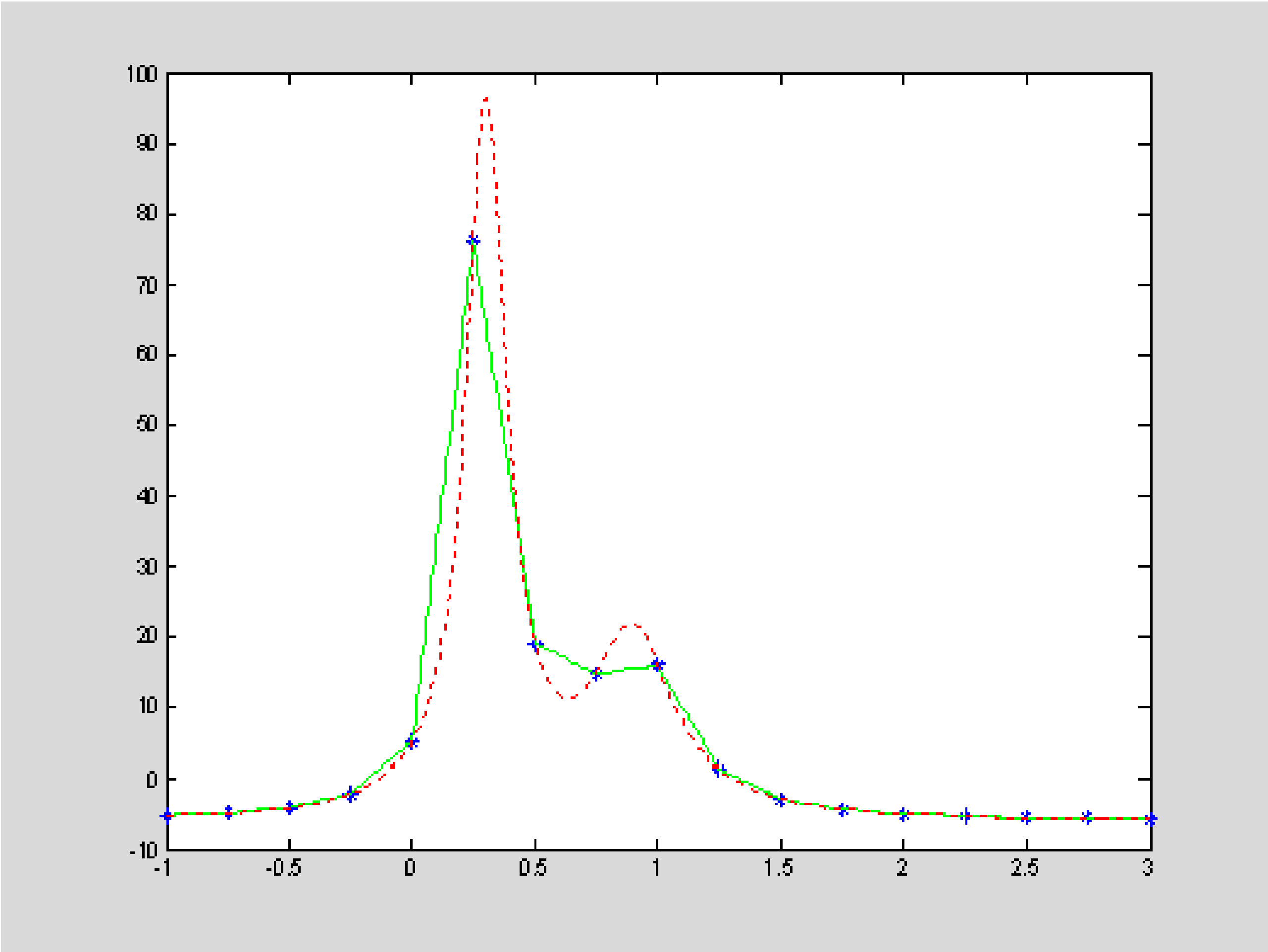
Lower order splines do not oscillate, but are not very smooth at the joints (knots) where they may change their characters (shape)

The derivative of the first order spline may not be continuous, the slope of the spline may change abruptly at the knots

The second derivative of the second order spline may not be continuous. The curvature of the quadratic spline may change abruptly at the knots

A better strategy is to use moderate order polynomials to construct spline and to impose smoothness conditions at the knots so that the resulting spline looks smooth but not oscillatory

# Linear Spline




Linear polynomial interpolation is not smooth

## Spline of Degree $k$

A function $S$ is a spline of degree $k$ if

1.) The domain of $S$ is an interval $[a,b]$

2.) $S, S', \ldots, S^{(k-1)}$, are all continuous functions on $[a,b]$

3.) There are points $t_i$ (the knots of $S$) such that $a = t_0 < t_1 < \ldots < t_n = b$ and such that $S$ is a polynomial of degree at most $k$ on each subinterval $[t_i, t_{i+1}]$

Smoothness is reflected by the order of the continuous derivative at the knots

If we want the approximating spline to have continuous $m$th derivative, we should choose a spline of degree at least $m + 1$

## Cubic Spline

For a spline of degree $(m + 1)$ at any interior knot $t_i$, we have

$$\lim_{x \to t_i^-} S^{(k)}(x) = \lim_{x \to t_i^+} S^{(k)}(x) \quad (0 \le k \le m)$$

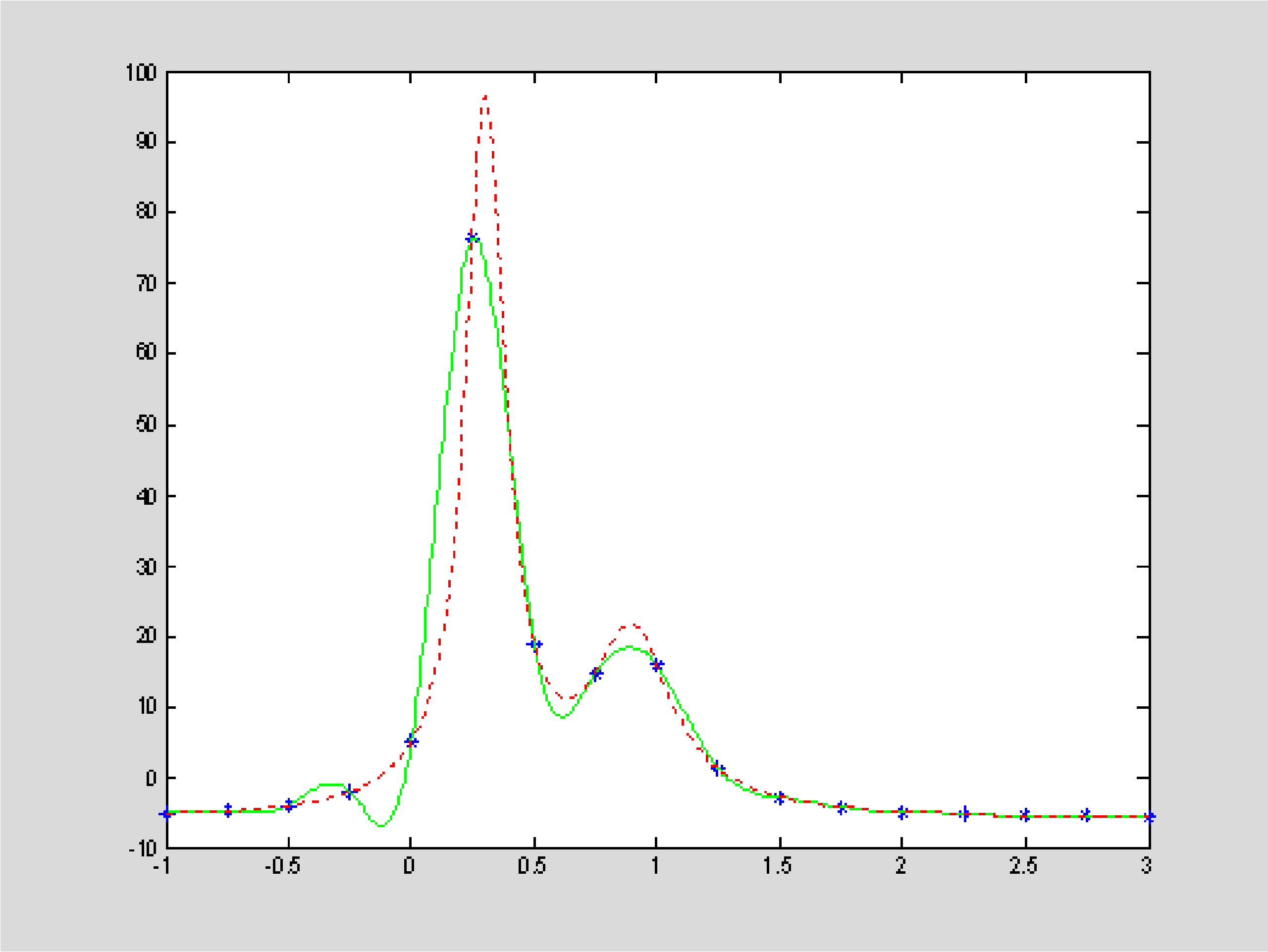
Since different polynomials are defined at different side of $t_i$, these two polynomials will be the same if they are at most degree $m$ and their zeroth to $m$th derivatives are equal

In most applications, a spline of degree $3$ is chosen (smooth and inexpensive to construct)

This spline has continuous first and second derivatives

Its graph looks smooth, although higher derivatives may be discontinuous

# Natural Cubic Spline




Cubic spline interpolation is more smooth

# Natural Cubic Spline

Given a table of data set

| $x$ | $t_0$ | $t_1$ | $\cdots$ | $t_n$ |
|---|---|---|---|---|
| $y$ | $y_0$ | $y_1$ | $\cdots$ | $y_n$ |

where the $t_i$'s are the knots in ascending order

We want to construct a function $S$ with $n$ piecewise cubic polynomials

$$S(x) = \begin{cases} S_0(x) & t_0 \le x \le t_1 \\ S_1(x) & t_1 \le x \le t_2 \\ \vdots & \vdots \\ S_{n-1}(x) & t_{n-1} \le x \le t_n \end{cases}$$

$S_i(x)$ is the cubic polynomial defined on the subinterval $[t_i, t_{i+1}]$, the cubic polynomial looks like

$$S_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \qquad 0 \le i \le n-1$$

We have a total of $4n$ unknowns to be determined

## Natural Cubic Spline

$S(x)$ must satisfy the interpolation conditions

$$S(t_i) = y_i \qquad (0 \le i \le n)$$

This will yield $2n$ conditions for $n$ intervals

$S(x)$, $S'(x)$ and $S''(x)$ have to satisfy the continuity conditions at the interior knots $t_1, t_2, \ldots, t_{n-1}$

$$\lim_{x \to t_i^-} S^{(k)}(x) = \lim_{x \to t_i^+} S^{(k)}(x) \qquad (k = 0,1,2)$$

This will yield $2(n-1)$ conditions

The remaining two (natural) conditions are imposed at the end knots as

$$S''(t_0) = S''(t_n) = 0$$

These particular conditions give the name of natural cubic spline. Other end knot conditions can also be imposed

# Computational Algorithm

Since $S''(x)$ is continuous, we define

$$z_i = S''(t_i) \qquad (0 \le i \le n)$$

with $z_0 = z_n = 0$. For the interval $[t_i, t_{i+1}]$ $S_i''(x)$ is continuous and takes the values of $z_i$ and $z_{i+1}$ at knots $t_i$ and $t_{i+1}$ (interpolation on interval $[t_i, t_{i+1}]$)

$$S_i''(x) = \frac{z_{i+1}}{h_i}(x - t_i) + \frac{z_i}{h_i}(t_{i+1} - x)$$

with $h_i = t_{i+1} - t_i$ for $0 \le i \le n - 1$

It can be verified that $S_i''(t_i) = z_i$, $S_i''(t_{i+1}) = z_{i+1}$ for the interpolation conditions of $S_i''(x)$

If we integrate $S_i''(x)$ twice, we get $S_i(x)$

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 + cx + d$$

with $c$ and $d$ being two constants

# Computational Algorithm (II)

The cubic polynomial can be rewritten as

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 + C_i(x - t_i) + D_i(t_{i+1} - x)$$

Since $S_i(x)$ must also interpolate at $t_i$ and $t_{i+1}$ we have $S_i(t_i) = y_i$ and $S_i(t_{i+1}) = y_{i+1}$, the constants $C_i$ and $D_i$ can be determined

We now have the polynomial as a function of the $z_i$'s

$$\begin{aligned} S_i(x) = {} & \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 \\ & + \left(\frac{y_{i+1}}{h_i} - \frac{h_i}{6} z_{i+1}\right)(x - t_i) \\ & + \left(\frac{y_i}{h_i} - \frac{h_i}{6} z_i\right)(t_{i+1} - x) \end{aligned}$$

## Computational Algorithm (III)

In order to impose the continuity conditions for $S'(x)$, we have

$$S_i'(x) = \frac{z_{i+1}}{2h_i}(x - t_i)^2 - \frac{z_i}{2h_i}(t_{i+1} - x)^2$$

$$+ \frac{y_{i+1}}{h_i} - \frac{h_i}{6} z_{i+1} - \frac{y_i}{h_i} - \frac{h_i}{6} z_i$$

We have at the knot $t_i$

$$S_i'(t_i) = -\frac{h_i}{6} z_{i+1} - \frac{h_i}{3} z_i + b_i$$

$$b_i = \frac{1}{h_i}(y_{i+1} - y_i)$$

and

$$S'_{i-1}(t_i) = -\frac{h_{i-1}}{6} z_{i-1} - \frac{h_{i-1}}{3} z_i + b_{i-1}$$

$$b_{i-1} = \frac{1}{h_{i-1}}(y_i - y_{i-1})$$

## Computational Algorithm (IV)

The continuity condition implies $S_i'(t_i) = S'_{i-1}(t_i)$, i.e.,

$$h_{i-1}z_{i-1} + 2(h_{i-1} + h_i)z_i + h_i z_{i+1} = 6(b_i - b_{i-1})$$

For $1 \le i \le n-1$. We have to solve a tridiagonal system of equations

$$z_0 = 0$$

$$h_{i-1}z_{i-1} + u_i z_i + h_i z_{i+1} = v_i \quad (1 \le i \le n-1)$$

$$z_n = 0$$

With

$$u_i = 2(h_{i-1} + h_i)$$

$$v_i = 6(b_i - b_{i-1})$$

Note that we have a tridiagonal system to solve

It can be shown that pivoting is not needed to solve the tridiagonal system, because the coefficient matrix is diagonally dominated

# Smoothness of Cubic Spline

The smoothness of an interpolation polynomial can be measured in some sense by the degree of fluctuation of its derivatives

The cubic spline can be shown as "smooth"

If $S$ is the natural cubic spline function that interpolates a twice continuously differentiable function $f$ at knots $a = t_0 < t_1 < \ldots < t_n = b$, then

$$\int_a^b [S''(x)]^2 \, dx \leq \int_a^b [f''(x)]^2 \, dx$$

If the graph of a function changes abruptly at some knot, we can construct two different natural cubic splines at different side of that knot to avoid forcing derivative continuity at a point where the function's derivative is not continuous

# Smoothness Proof

We define

$$g(x) = f(x) - S(x)$$

It follows that

$$g(t_i) = 0 \quad \text{for} \quad 0 \le i \le n$$

because $t_i$ are the knot points
Furthermore

$$f''(x) = S''(x) + g''(x)$$

And

$$\int_a^b (f'')^2 dx = \int_a^b (S'')^2 dx + \int_a^b (g'')^2 dx + 2\int_a^b S'' g'' dx$$

# Smoothness Proof

Look at the last term and use integration by parts

$$\int_a^b S''g''dx = S''g'|_a^b - \int_a^b S'''g'dx = -\int_a^b S'''g'dx$$

We used the facts of natural cubic spline that

$$S''(a) = 0 \quad \text{and} \quad S''(b) = 0$$

Now, we have

$$\int_a^b S'''g'dx = \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} S'''g'dx$$

Since S is a cubic polynomial in each interval [*ti*, *ti+1*], its third derivative is a constant, say ci, so

$$\int_a^b S'''g'dx = \sum_{i=0}^{n-1} c_i \int_{t_i}^{t_{i+1}} g'dx = \sum_{i=0}^{n-1} c_i [g(t_{i+1}) - g(t_i)] = 0$$

## Smoothness Proof

So from

$$\int_a^b (f'')^2 dx = \int_a^b (S'')^2 dx + \int_a^b (g'')^2 dx + 2\int_a^b S'' g'' dx$$

And the fact that

$$\int_a^b S'' g'' dx = 0$$

We have

$$\int_a^b (f'')^2 dx = \int_a^b (S'')^2 dx + \int_a^b (g'')^2 dx \geq \int_a^b (S'')^2 dx$$

Thus, the smoothness of the cubic spline is proved