Computational methods and numerical algorithms: Lecture 4

Wangtao Lu
Zhejiang University

September 29, 2020

Contents

Wangtao Lu
Zhejiang University

3.1 Data and Interpolation Functions

A function $y = P(x)$ is said to **interpolate** a set of data points $\{(x_i, y_i)\}_{i=1}^{n}$ if it passes through those points, i.e., $y_i = P(x_i)$.
Example 1. A parabola that interpolates $(0, 1)$, $(2, 2)$ and $(3, 4)$ is

$$y = P(x) = \frac{1}{2}x^2 - \frac{1}{2}x + 1.$$

Candidates of the **interpolation function** $P(x)$: all preliminary functions, such as:

- Polynomials: $1 + x, x^2 + 2x^3, \ldots$
- Rational functions: $\frac{1+x}{1+x^2}, \ldots$
- Trigonometric functions: $\sin(x), \cos(x), \tan(x), \cot(x), \ldots$
- Other functions: $\sinh(x), \ln(x), \sqrt{x}, e^x, \ldots$
- Composition of the above functions

Polynomial Interpolations: Lagrange approach

Assume that n data points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ are given with distinct $\{x_i\}_{i=1}^n$, and we would like to find an interpolating polynomial $P(x)$. Lagrange approach: For $1 \leq j \leq n$, find a polynomial $L_j(x)$ of the smallest degree such that

$$L_j(x_j) = 1, \quad L_j(x_i) = 0, \forall i \neq j.$$

Then, a possible interpolating polynomial is

$$P(x) = \sum_{i=1}^{n} y_i L_i(x),$$

which is the **Lagrange interpolating polynomial** of degree $n-1$ or less, and $\{L_j(x)\}_{j=1}^n$ are called the basis functions.

Polynomial Interpolations: Lagrange approach

Assume that n data points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ are given with distinct $\{x_i\}_{i=1}^n$, and we would like to find an interpolating polynomial $P(x)$. Lagrange approach: For $1 \leq j \leq n$, find a polynomial $L_j(x)$ of the smallest degree such that

$$L_j(x_j) = 1, \quad L_j(x_i) = 0, \forall i \neq j.$$

It is easy to derive that

$$L_j(x) = a_j \prod_{i=1, i\neq j}^{n} (x - x_i), \qquad .$$

Then, a possible interpolating polynomial is

$$P(x) = \sum_{i=1}^{n} y_i L_i(x),$$

which is the **Lagrange interpolating polynomial** of degree $n-1$ or less, and $\{L_j(x)\}_{j=1}^n$ are called the basis functions.

Polynomial Interpolations: Lagrange approach

Assume that n data points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ are given with distinct $\{x_i\}_{i=1}^n$, and we would like to find an interpolating polynomial $P(x)$. Lagrange approach: For $1 \leq j \leq n$, find a polynomial $L_j(x)$ of the smallest degree such that

$$L_j(x_j) = 1, \quad L_j(x_i) = 0, \forall i \neq j.$$

It is easy to derive that

$$L_j(x) = a_j \prod_{i=1, i\neq j}^{n} (x - x_i), \quad a_j = \left(\prod_{i=1, i\neq j}^{n} (x_j - x_i) \right)^{-1}.$$

Then, a possible interpolating polynomial is

$$P(x) = \sum_{i=1}^{n} y_i L_i(x),$$

which is the **Lagrange interpolating polynomial** of degree $n-1$ or less, and $\{L_j(x)\}_{j=1}^n$ are called the basis functions.

Example 2. Find the Lagrange interpolating polynomial for the data points $(0,1)$, $(2,2)$ and $(3,4)$.

Sol. First, we find

$$L_1(x) = \frac{(x-2)(x-3)}{(0-2)(0-3)},$$
$$L_2(x) = \frac{(x-0)(x-3)}{(2-0)(2-3)},$$
$$L_3(x) = \frac{(x-0)(x-2)}{(3-0)(3-2)}.$$

Then, we get the Lagrange interpolating polynomial

$$P(x) = 1 \cdot L_1(x) + 2 \cdot L_2(x) + 3 \cdot L_3(x) = \frac{1}{2}x^2 - \frac{1}{2}x + 1.$$

Main Theorem of Polynomial Interpolation

Theorem. Let $(x_1, y_1), \ldots, (x_n, y_n)$ be n points in the plane with distinct x_i. Then there exists one and only one polynomial P of degree $n-1$ or less that satisfies $P(x_i) = y_i$ for $i = 1, \ldots, n$.

Proof. It is easy to see that the Lagrange interpolating polynomial meets the requirement which proves the existence of P.

To show that such an interpolating polynomial is unique, we suppose that there are two distinct polynomials P_1 and P_2 of degrees less than or equal to $n-1$ such that $P_1(x_i) = P_2(x_i) = y_i$ for $i = 1, \ldots, n$. Then, polynomial $P_1 - P_2$ has n distinct roots $x_1, \ldots, x_n$ since

$$P_1(x_i) - P_2(x_i) = 0.$$

But $P_1 - P_2$ has at most $n-1$ roots according to the Fundamental Theorem of Algebra since the degree of $P_1 - P_2$ is less than n. This is a contradiction.

Consequently, the Lagrange interpolating polynomial is the unique polynomial of degree $n-1$ or less that interpolates the n points.

Degree strictly less than $n-1$

Wangtao Lu
Zhejiang University

Example 3. Find the polynomial of degree 3 or less that interpolates $(0,2),(1,1),(2,0)$ and $(3,-1)$.

Sol. The Lagrange interpolating polynomial is

$$\begin{aligned} P(x) =& 2\frac{(x-1)(x-2)(x-3)}{(0-1)(0-2)(0-3)} + 1\frac{(x-0)(x-2)(x-3)}{(1-0)(1-2)(1-3)} \\ &+ 0\frac{(x-0)(x-1)(x-3)}{(2-0)(2-1)(2-3)} + (-1)\frac{(x-0)(x-1)(x-2)}{(3-0)(3-1)(3-2)} \\ =& -x+2. \end{aligned}$$

Remark. The Main Theorem of Polynomial Interpolation indicates for this example that no polynomial of degree 2 or 3 could interpolate the four collinear points. This also reflects that no parabola or cubic curves could pass through four collinear points.

Newton's divided differences

Denote by $f[x_1, \dots, x_n]$ the coefficient of the x^{n-1} term in the (unique) polynomial that interpolates $(x_1, f(x_1)), \dots, (x_n, f(x_n))$.

Theorem. The Lagrange interpolating polynomial $P(x)$ of degree $n-1$ or less that interpolates $\{(x_j, f(x_j))\}_{j=1}^n$ takes the following form

$$\begin{aligned} P(x) =& f[x_1] + f[x_1, x_2](x - x_1) + f[x_1, x_2, x_3](x - x_1)(x - x_2) \\ &+ \cdots + f[x_1, \dots, x_n](x - x_1)(x - x_2) \dots (x - x_{n-1}). \end{aligned}$$

Proof. We prove the theorem by induction. Case $n = 1$ is easy to verify. We now assume that the above theorem holds for $n = k$. When $n = k + 1$, we see that

$$P_k(x) = f[x_1] + \cdots + f[x_1, \dots, x_k](x - x_1)(x - x_2) \dots (x - x_{k-1}),$$

should interpolate $\{(x_j, f(x_j))\}_{j=1}^k$. Thus,

$$P_{k+1}(x) = P_k(x) + a_k(x - x_1)(x - x_2) \dots (x - x_k),$$

passes through $\{(x_j, f(x_j))\}_{j=1}^k$ by the assumption for $n = k$. If $P_{k+1}(x)$ passes $(x_{k+1}, f(x_{k+1}))$, then $P_{k+1}(x)$ interpolates $\{(x_j, f(x_j))\}_{j=1}^{k+1}$ and is the only polynomial of degree k or less. Consequently,

$$f[x_1, \dots, x_k, x_{k+1}] = a_k,$$

since the coefficient of x^k-term of $P_{k+1}(x)$ is a_k.

Newton's Difference Formula

$$
\begin{aligned}
P(x) = & f[x_1] + f[x_1, x_2](x - x_1) + \cdots + f[x_1, \ldots, x_{n-1}](x - x_1) \cdots (x - x_{n-2}) \\
& + f[x_1, \ldots, x_n](x - x_1) \ldots (x - x_{n-1}),
\end{aligned} \tag{1}
$$

is an alternative form of Lagrange interpolating polynomial, and is called the **Newton's Difference Formula**.

Clearly, $P(x_1) = f(x_1)$ gives $f[x_1] = f(x_1)$. But what is $f[x_1, \ldots, x_k]$ for $k = 2, \ldots, n$?

One easily verifies that Newton's Difference Formula for the set $\{(x_2, f(x_2)), \ldots, (x_n, f(x_n)), (x_1, f(x_1))\}$ becomes

$$
\begin{aligned}
P(x) = & f[x_2] + f[x_2, x_3](x - x_2) + \cdots + f[x_2, \ldots, x_n](x - x_2) \cdots (x - x_{n-1}) \\
& + f[x_2, \ldots, x_n, x_1](x - x_2) \ldots (x - x_n).
\end{aligned} \tag{2}
$$

Newton's Difference Formula

Wangtao Lu
Zhejiang University

$$
\begin{aligned}
P(x) =& f[x_1] + f[x_1, x_2](x - x_1) + \cdots + f[x_1, \ldots, x_{n-1}](x - x_1)\cdots(x - x_{n-2}) \\
&+ f[x_1, \ldots, x_n](x - x_1)\ldots(x - x_{n-1}), \qquad (1)
\end{aligned}
$$

is an alternative form of Lagrange interpolating polynomial, and is called the **Newton's Difference Formula**.

Clearly, $P(x_1) = f(x_1)$ gives $f[x_1] = f(x_1)$. But what is $f[x_1, \ldots, x_k]$ for $k = 2, \ldots, n$?

One easily verifies that Newton's Difference Formula for the set $\{(x_2, f(x_2)), \ldots, (x_n, f(x_n)), (x_1, f(x_1))\}$ becomes

$$
\begin{aligned}
P(x) =& f[x_2] + f[x_2, x_3](x - x_2) + \cdots + f[x_2, \ldots, x_n](x - x_2)\cdots(x - x_{n-1}) \\
&+ f[x_2, \ldots, x_n, x_1](x - x_2)\ldots(x - x_n). \qquad (2)
\end{aligned}
$$

Comparing x^{n-1}-terms in (1) and (2),

$$
f[x_1, \ldots, x_n] = f[x_2, \ldots, x_n, x_1].
$$

Newton's Difference Formula

$$\begin{aligned} P(x) =& f[x_1] + f[x_1, x_2](x - x_1) + \cdots + f[x_1, \ldots, x_{n-1}](x - x_1)\cdots(x - x_{n-2}) \\ &+ f[x_1, \ldots, x_n](x - x_1)\ldots(x - x_{n-1}), \end{aligned} \quad (1)$$

is an alternative form of Lagrange interpolating polynomial, and is called the **Newton's Difference Formula**.
Clearly, $P(x_1) = f(x_1)$ gives $f[x_1] = f(x_1)$. But what is $f[x_1, \ldots, x_k]$ for $k = 2, \ldots, n$?
One easily verifies that Newton's Difference Formula for the set $\{(x_2, f(x_2)), \ldots, (x_n, f(x_n)), (x_1, f(x_1))\}$ becomes

$$\begin{aligned} P(x) =& f[x_2] + f[x_2, x_3](x - x_2) + \cdots + f[x_2, \ldots, x_n](x - x_2)\cdots(x - x_{n-1}) \\ &+ f[x_2, \ldots, x_n, x_1](x - x_2)\ldots(x - x_n). \end{aligned} \quad (2)$$

Comparing x^{n-1}-terms in (1) and (2),

$$f[x_1, \ldots, x_n] = f[x_2, \ldots, x_n, x_1].$$

Now considering $(n-2)$-th order derivative of (1) and (2),

Newton's Difference Formula

$$
\begin{aligned}
P(x) =& f[x_1] + f[x_1, x_2](x - x_1) + \cdots + f[x_1, \ldots, x_{n-1}](x - x_1)\cdots(x - x_{n-2}) \\
&+ f[x_1, \ldots, x_n](x - x_1)\ldots(x - x_{n-1}), \qquad (1)
\end{aligned}
$$

is an alternative form of Lagrange interpolating polynomial, and is called the **Newton's Difference Formula**.

Clearly, $P(x_1) = f(x_1)$ gives $f[x_1] = f(x_1)$. But what is $f[x_1, \ldots, x_k]$ for $k = 2, \ldots, n$?

One easily verifies that Newton's Difference Formula for the set $\{(x_2, f(x_2)), \ldots, (x_n, f(x_n)), (x_1, f(x_1))\}$ becomes

$$
\begin{aligned}
P(x) =& f[x_2] + f[x_2, x_3](x - x_2) + \cdots + f[x_2, \ldots, x_n](x - x_2)\cdots(x - x_{n-1}) \\
&+ f[x_2, \ldots, x_n, x_1](x - x_2)\ldots(x - x_n). \qquad (2)
\end{aligned}
$$

Comparing x^{n-1}-terms in (1) and (2),

$$f[x_1, \ldots, x_n] = f[x_2, \ldots, x_n, x_1].$$

Now considering $(n-2)$-th order derivative of (1) and (2),

$$
\begin{aligned}
&f[x_1, \ldots, x_{n-1}](n-2)! + f[x_1, \ldots, x_n](n-2)!\sum_{i=1}^{n-1}(x - x_i) \\
=&f[x_2, \ldots, x_n](n-2)! + f[x_2, \ldots, x_n, x_1](n-2)!\sum_{i=2}^{n}(x - x_i).
\end{aligned}
$$

Newton's Difference Formula

Wangtao Lu
Zhejiang University

$$
\begin{aligned}
P(x) =& f[x_1] + f[x_1,x_2](x-x_1) + \cdots + f[x_1,\ldots,x_{n-1}](x-x_1)\cdots(x-x_{n-2}) \\
&+ f[x_1,\ldots,x_n](x-x_1)\ldots(x-x_{n-1}),
\end{aligned} \tag{1}
$$

is an alternative form of Lagrange interpolating polynomial, and is called the **Newton's Difference Formula**.

Clearly, $P(x_1) = f(x_1)$ gives $f[x_1] = f(x_1)$. But what is $f[x_1,\ldots,x_k]$ for $k = 2,\ldots,n$?

One easily verifies that Newton's Difference Formula for the set $\{(x_2, f(x_2)),\ldots,(x_n, f(x_n)),(x_1, f(x_1))\}$ becomes

$$
\begin{aligned}
P(x) =& f[x_2] + f[x_2,x_3](x-x_2) + \cdots + f[x_2,\ldots,x_n](x-x_2)\cdots(x-x_{n-1}) \\
&+ f[x_2,\ldots,x_n,x_1](x-x_2)\ldots(x-x_n).
\end{aligned} \tag{2}
$$

Comparing x^{n-1}-terms in (1) and (2),

$$
f[x_1,\ldots,x_n] = f[x_2,\ldots,x_n,x_1].
$$

Now considering $(n-2)$-th order derivative of (1) and (2),

$$
f[x_1,\ldots,x_n] = \frac{f[x_2,\ldots,x_n] - f[x_1,\ldots,x_{n-1}]}{x_n - x_1}, \quad n \geq 2.
$$

Newton's Divided Differences

In summary, we get the following formula

$$f[x_1] = f(x_1),$$
$$f[x_1, \dots, x_n] = \frac{f[x_2, \dots, x_n] - f[x_1, \dots, x_{n-1}]}{x_n - x_1}, \quad n \geq 2.$$

To make use of Newton's Divided Difference Formula,

$$P(x) = f[x_1] + f[x_1, x_2](x - x_1) + \cdots + f[x_1, \dots, x_{n-1}](x - x_1) \cdots (x - x_{n-2}) + f[x_1, \dots, x_n](x - x_1) \dots (x - x_{n-1}),$$

we need the following divided differences,

$$f[x_k] = \qquad , 1 \leq k \leq n,$$

$$f[x_k, \dots, x_{k+p}] = \qquad\qquad , \quad 1 \leq p \leq n - k.$$

Computational Complexity:

$$\mathcal{O}(n^2).$$

Newton's Divided Differences

Wangtao Lu
Zhejiang University

In summary, we get the following formula

$$\begin{aligned} f[x_1] &= f(x_1), \\ f[x_1, \ldots, x_n] &= \frac{f[x_2, \ldots, x_n] - f[x_1, \ldots, x_{n-1}]}{x_n - x_1}, \quad n \geq 2. \end{aligned}$$

To make use of Newton's Divided Difference Formula,

$$\begin{aligned} P(x) =& f[x_1] + f[x_1, x_2](x - x_1) + \cdots + f[x_1, \ldots, x_{n-1}](x - x_1) \cdots (x - x_{n-2}) \\ &+ f[x_1, \ldots, x_n](x - x_1) \ldots (x - x_{n-1}), \end{aligned}$$

we need the following divided differences,

$$\begin{aligned} f[x_k] &= f(x_k), 1 \leq k \leq n, \\ f[x_k, \ldots, x_{k+p}] &= \frac{f[x_{k+1}, \ldots, x_{k+p}] - f[x_k, \ldots, x_{k+1}]}{x_{k+p} - x_k}, \quad 1 \leq p \leq n - k. \end{aligned}$$

Computational Complexity:

$$\mathcal{O}(n^2).$$

Divided Difference Triangle

Consider interpolating three points $(x_1, f(x_1)), (x_2, f(x_2))$, and $(x_3, f(x_3))$. We can build up the following table

$$
\begin{array}{c|ccc}
x_1 & f[x_1] & & \\
 & & f[x_1, x_2] & \\
x_2 & f[x_2] & & f[x_1, x_2, x_3] \\
 & & f[x_2, x_3] & \\
x_3 & f[x_3] & &
\end{array}
$$

Then, the Newton's divided difference formula indicates the interpolating polynomial is

$$P(x) = f[x_1] + f[x_1, x_2](x - x_1) + f[x_1, x_2, x_3](x - x_1)(x - x_2).$$

Example 3.(a).Find the interpolating polynomial passing through $(0,1),(2,2),(3,4)$. (b). Add one more point $(1,2)$ in (a).

(a).Newton's divided differences

$$
\begin{array}{c|cccc}
0 & 1 & & \\
 & & 1/2 & \\
2 & 2 & & 1/2 \\
 & & 2 & \\
3 & 4 & &
\end{array}
$$

$P(x) = 1 + \frac{1}{2}(x-0) + \frac{1}{2}(x-0)(x-2)$.

(a) The Lagrange basis functions

$$L_1(x) = \frac{(x-3)(x-2)}{(0-3)(0-2)},$$
$$L_2(x) = \frac{(x-3)(x-0)}{(2-3)(2-0)},$$
$$L_3(x) = \frac{(x-2)(x-0)}{(3-2)(3-0)},$$

$P(x) = L_1(x) + 2L_2(x) + 4L_3(x)$.

(b). Newton's divided differences

$$
\begin{array}{c|ccccccc}
0 & 1 & & & \\
 & & 1/2 & & \\
2 & 2 & & 1/2 & \\
 & & 2 & & 1/2 \\
3 & 4 & & 1 & \\
 & & 1 & & \\
1 & 2 & & &
\end{array}
$$

$P(x) = 1 + \frac{1}{2}(x-0) + \frac{1}{2}(x-0)(x-2) + \frac{1}{2}(x-0)(x-2)(x-3)$.

(b) The Lagrange basis functions

$$L_1(x) = \frac{(x-1)(x-3)(x-2)}{(0-1)(0-3)(0-2)},$$
$$L_2(x) = \frac{(x-1)(x-3)(x-0)}{(2-1)(2-3)(2-0)},$$
$$L_3(x) = \frac{(x-1)(x-2)(x-0)}{(3-1)(3-2)(3-0)},$$
$$L_4(x) = \frac{(x-3)(x-2)(x-0)}{(1-3)(1-2)(1-0)},$$

$P(x) =$
$L_1(x) + 2L_2(x) + 4L_3(x) + 2L_4(x)$

Newton vs. Lagrange

From above, we can see at least two advantages of Newton's divided difference formula over Lagrange's interpolating polynomial.

1. New data points can be easily added in the sense that only a few work needs to be done for each new added point.
2. Nested Method can be directly applied to compute the resulting interpolating polynomial at any given point x.

Wangtao Lu
Zhejiang University

How many degree d polynomials pass through n points?

The answer is **at most one** when $d \leq n-1$. If $d \geq n$, then the answer is infinitely many.

Example 5. How many polynomials of each degree $0 \leq d \leq 5$ pass through the points $(-1,-5),(0,-1),(2,1)$, and $(3,11)$?

Sol. Newton's Divided Difference

$$
\begin{array}{r|cccc}
-1 & -5 & & & \\
 & & 4 & & \\
0 & -1 & & -1 & \\
 & & 1 & & 1 \\
2 & 1 & & 3 & \\
 & & 10 & & \\
3 & 11 & & &
\end{array}
$$

The unique polynomial of degree $4-1$ or less is

$$P_3(x) = -5 + 4(x+1) - (x+1)x + (x+1)x(x-2).$$

When $0 \leq d \leq 2$, the answer is 0. When $d = 3$, the answer is 1. When $d = 4$, the polynomials interpolating the above four points must take the following form

$$P_4(x) = -5+4(x+1)-(x+1)x+(x+1)x(x-2)+c_1(x+1)x(x-2)(x-3).$$

When $d = 5$,

$$P_5(x) = -5+4(x+1)-(x+1)x+(x+1)x(x-2)+(c_2x+c_1)(x+1)x(x-2)(x-3).$$

Interpolating error formula

Theorem. Suppose $f(x)$ is n-th continuously differentiable for $x \in [a, b]$. Assume that $P(x)$ is the (degree $n-1$ or less) interpolating polynomial passes through $(x_1, f(x_1)), \ldots, (x_n, f(x_n))$. The interpolating error is

$$\max_{a \le x \le b} |f(x) - P(x)| = \max_{a \le x \le b} \left| \frac{(x-x_1)\cdots(x-x_n)}{n!} f^{(n)}(\xi) \right|,$$

where ξ lies between the smallest and largest of the numbers $x, x_1, \ldots, x_n$.

Proof. Using Newton's divided difference formula,

$$\begin{aligned} P(t) =& f[x_1] + f[x_1, x_2](t - x_1) + f[x_1, x_2, x_3](t - x_1)(t - x_2) \\ & + \cdots + f[x_1, \ldots, x_n](t - x_1)(t - x_2)\ldots(t - x_{n-1}). \end{aligned}$$

Then, considering add a new point $(x, f(x))$, we get

$$h(t) = P(t) + f[x_1, \ldots, x_n, x](t - x_1)\ldots(t - x_n).$$

At $t = x$,

$$f(x) = h(x) = P(x) + f[x_1, \dots, x_n, x](x - x_1)\dots(x - x_n).$$

Then,

$$g(t) = f(t) - P(t) - f[x_1, \dots, x_n, x](t - x_1)\dots(t - x_n),$$

has $n+1$ distinct roots $x_1, \dots, x_n, x$. Using Roll's Theorem, we see that $g'(t)$ has n distinct roots. Roll's Theorem again indicates that $g''(t)$ has $n-1$ distinct roots. Repearting this procedure, we conclude that $g^{(n)}(t)$ has one root ξ between the smallest and largest number of $x_1, \dots, x_n, x$. Consequently,

$$f^{(n)}(\xi) - P^{(n)}(\xi) - f[x_1, \dots, x_n, x]n! = 0 \to f[x_1, \dots, x_n, x] = \frac{f^{(n)}(\xi)}{n!}.$$

Thus,

$$f(x) - P(x) = \frac{f^{(n)}(\xi)}{n!}(x - x_1)\dots(x - x_n).$$

Examples

Example 1. Interpolate the function $f(x) = \sin x$ at 4 equally spaced points on $[0, \pi/2]$, and analyze the interpolation error at $x = 1, 0.2$.

Sol. Newton's divided difference table:

$$
\begin{array}{c|cccc}
0 & 0.0000 & & & \\
 & & 0.9549 & & \\
\pi/6 & 0.5000 & & -0.2443 & \\
 & & 0.6990 & & -0.1139 \\
2\pi/6 & 0.8660 & & -0.4232 & \\
 & & 0.2559 & & \\
3\pi/6 & 1.0000 & & &
\end{array}
$$

The degree 3 interpolating polynomial is therefore

$$P_3(x) = 0 + 0.9549x - 0.2443x(x - \pi/6) - 0.1139x(x - \pi/6)(x - \pi/3).$$

By the interpolation error formula,

$$\sin x - P_3(x) = \frac{(x-0)(x-\pi/6)(x-\pi/3)(x-\pi/2)}{4!} f^{(4)}(\xi).$$

Thus,

$$|\sin x - P_3(x)| \leq \left| \frac{(x-0)(x-\pi/6)(x-\pi/3)(x-\pi/2)}{24} \right|$$

When $x = 1$,

$$|\sin 1 - P_3(1)| \leq \left|\frac{(1-0)(1-\pi/6)(1-\pi/3)(1-\pi/2)}{24}\right| \approx 0.0005348,$$

and in fact

$$|\sin 1 - P_3(1)| \approx 0.0004.$$

When $x = 0.2$,

$$|\sin 0.2 - P_3(0.2)| \leq \left|\frac{(0.2-0)(0.2-\pi/6)(0.2-\pi/3)(0.2-\pi/2)}{24}\right| \approx 0.00313,$$

and in fact,

$$|\sin 0.2 - P_3(0.2)| \approx 0.00189$$

Runge Phenomenon

Example 4 Interpolate $f(x) = \frac{1}{1+12x^2}$ at evenly spaced points in $[-1, 1]$.
Sol.

See the textbook for results.

3.3 How to reduce the interpolation error?

Recall the interpolation error

$$\max_{a \le x \le b} \left| \frac{(x - x_1)(x - x_2) \dots (x - x_n)}{n!} f^{(n)}(c) \right|.$$

What we can do only is to choose proper base points to reduce the above interpolation error!
Runge's Phenomena tells us that Equally spaced base points are not a good choice!
Example 1: In $[-1, 1]$, we plot $(x - x_1) \dots (x - x_n)$ in MATLAB using the following two sets of base points:

1. Evenly spaced points $x_i = -1 + 2(i - 1)/(n - 1), i = 1, \dots, n$.
2. Unevenly spaced points $x_i = \cos \frac{(2i-1)\pi}{2n}, i = 1, \dots, n$.

Wangtao Lu
Zhejiang University

Chebyshev polynomial

The unevenly spaced points

$$x_i = \cos\frac{(2i-1)\pi}{2n}, \quad i = 1, \ldots, n,$$

are in fact the roots of

$$T_n = \cos(n \arccos x), \quad -1 \le x \le 1.$$

which we call the degree n Cheyshev polynomial.
Why it is a polynomial? Use the following relation:

$$2\cos\alpha\cos\beta = \cos(\alpha+\beta) + \cos(\alpha-\beta).$$

We take

$$\alpha = \qquad\qquad\qquad , \beta = \qquad\qquad ,$$

and get the following recursion relation for the Chebyshev polynomials.

Chebyshev polynomial

The unevenly spaced points

$$x_i = \cos\frac{(2i-1)\pi}{2n}, \quad i = 1, \ldots, n,$$

are in fact the roots of

$$T_n = \cos(n \arccos x), \quad -1 \leq x \leq 1.$$

which we call the degree n Cheyshev polynomial.
Why it is a polynomial? Use the following relation:

$$2\cos\alpha\cos\beta = \cos(\alpha+\beta) + \cos(\alpha-\beta).$$

We take

$$\alpha = (n-1)\cdot\arccos x, \beta = 1\cdot\arccos x,$$

and get the following recursion relation for the Chebyshev polynomials.

Chebyshev polynomial

Wangtao Lu
Zhejiang University

The unevenly spaced points

$$x_i = \cos\frac{(2i-1)\pi}{2n}, \quad i = 1, \ldots, n,$$

are in fact the roots of

$$T_n = \cos(n \arccos x), \quad -1 \leq x \leq 1.$$

which we call the degree n Cheyshev polynomial.
Why it is a polynomial? Use the following relation:

$$2\cos\alpha\cos\beta = \cos(\alpha+\beta) + \cos(\alpha-\beta).$$

We take

$$\alpha = (n-1)\cdot\arccos x, \beta = 1\cdot\arccos x,$$

and get the following recursion relation for the Chebyshev polynomials.

$$2xT_{n-1}(x) = T_n(x) + T_{n-2}(x).$$

Properties of Chebyshev Polynomials

Wangtao Lu
Zhejiang University

Prop. 1 The n degree Chebyshev polynomial $T_n(x) = \cos(n \arccos x)$ has and only has n roots in $[-1, 1]$:

$$x_i = \cos \frac{(2i-1)\pi}{2n}, \quad i = 1, \ldots, n,$$

and satisfies the following recursion relation

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n \geq 2.$$

The first few Cheybshev polynomials are

$$\begin{aligned} T_0(x) &= 1, \quad T_1(x) = x, \\ T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x. \end{aligned}$$

Prop. 2 $\{x_i\}_{i=1}^n$ satisfies

$$(x - x_1) \ldots (x - x_n) = \underline{\qquad} T_n(x),$$

since the leading coefficient of T_n is ____.
Prop. 3 The maximum absolute value of $T_n(x)$ for $-1 \leq x \leq 1$ is   .

Properties of Chebyshev Polynomials

Prop. 1 The n degree Chebyshev polynomial $T_n(x) = \cos(n \arccos x)$ has and only has n roots in $[-1, 1]$:

$$x_i = \cos \frac{(2i-1)\pi}{2n}, \quad i = 1, \ldots, n,$$

and satisfies the following recursion relation

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n \geq 2.$$

The first few Cheybshev polynomials are

$$\begin{aligned} T_0(x) &= 1, \quad T_1(x) = x, \\ T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x. \end{aligned}$$

Prop. 2 $\{x_i\}_{i=1}^n$ satisfies

$$(x - x_1) \ldots (x - x_n) = \underline{\qquad} T_n(x),$$

since the leading coefficient of T_n is $\underline{2^{n-1}}$.
Prop. 3 The maximum absolute value of $T_n(x)$ for $-1 \leq x \leq 1$ is __ .

Properties of Chebyshev Polynomials

Prop. 1 The n degree Chebyshev polynomial $T_n(x) = \cos(n \arccos x)$ has and only has n roots in $[-1, 1]$:

$$x_i = \cos \frac{(2i-1)\pi}{2n}, \quad i = 1, \dots, n,$$

and satisfies the following recursion relation

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n \geq 2.$$

The first few Cheybshev polynomials are

$$\begin{aligned} T_0(x) &= 1, \quad T_1(x) = x, \\ T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x. \end{aligned}$$

Prop. 2 $\{x_i\}_{i=1}^n$ satisfies

$$(x - x_1) \dots (x - x_n) = \underline{\frac{1}{2^{n-1}}} T_n(x),$$

since the leading coefficient of T_n is $\underline{2^{n-1}}$.
Prop. 3 The maximum absolute value of $T_n(x)$ for $-1 \leq x \leq 1$ is .

Properties of Chebyshev Polynomials

Prop. 1 The n degree Chebyshev polynomial $T_n(x) = \cos(n \arccos x)$ has and only has n roots in $[-1, 1]$:

$$x_i = \cos \frac{(2i-1)\pi}{2n}, \quad i = 1, \ldots, n,$$

and satisfies the following recursion relation

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n \geq 2.$$

The first few Cheybshev polynomials are

$$\begin{aligned} T_0(x) &= 1, \quad T_1(x) = x, \\ T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x. \end{aligned}$$

Prop. 2 $\{x_i\}_{i=1}^n$ satisfies

$$(x - x_1) \ldots (x - x_n) = \frac{1}{\underline{2^{n-1}}} T_n(x),$$

since the leading coefficient of T_n is $\underline{2^{n-1}}$.
Prop. 3 The maximum absolute value of $T_n(x)$ for $-1 \leq x \leq 1$ is 1.

Chebyshev interpolating polynomial

Wangtao Lu
Zhejiang University

Using Chebyshev polynomial T_{n+1}'s roots as the base points, the interpolating polynomial is the so-called degree Chebyshev interpolating polynomial.

Example 2. Find a worst-case error bound for the difference on $[-1,1]$ between $f(x)=e^x$ and the degree 4 Chebyshev interpolating polynomial.

Sol. The interpolation error formula gives the following error upper bound

$$\max_{x\in -1\le 1}\left|\frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)(x-x_5)}{5!}f^{(5)}(c)\right|$$

For Chebyshev base points $x_i=\cos\frac{(2i-1)\pi}{10}, i=1,\dots,5$

$$\left|\frac{T_5(x)}{5!2^{5-1}}e^c\right| \le \frac{e}{5!2^4}\approx 0.00142,$$

For evenly spaced based points:

$$\left|\frac{(x+1)(x+0.5)x(x-0.5)(x-1)}{5!2^{5-1}}e^c\right| \le \frac{0.1135e}{5!}\approx 0.0026,$$

Chebyshev interpolating polynomial

Using Chebyshev polynomial T_{n+1}'s roots as the base points, the interpolating polynomial is the so-called degree n Chebyshev interpolating polynomial.

Example 2. Find a worst-case error bound for the difference on $[-1, 1]$ between $f(x) = e^x$ and the degree 4 Chebyshev interpolating polynomial.

Sol. The interpolation error formula gives the following error upper bound

$$\max_{x\in -1\leq 1}\left|\frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)(x-x_5)}{5!}f^{(5)}(c)\right|$$

For Chebyshev base points $x_i = \cos\frac{(2i-1)\pi}{10}, i = 1, \ldots, 5$

$$\left|\frac{T_5(x)}{5!2^{5-1}}e^c\right| \leq \frac{e}{5!2^4} \approx 0.00142,$$

For evenly spaced based points:

$$\left|\frac{(x+1)(x+0.5)x(x-0.5)(x-1)}{5!2^{5-1}}e^c\right|$$

$$\leq \frac{0.1135e}{5!} \approx 0.0026,$$

Chebyshev Theorem

Prop. 4 $T_n(1) = \_$ and $T_n(-1) = \underline{\qquad}$.
Prop. 5 $T_n(x)$ alternates between -1 and 1 a total of $n+1$ times.
This happens at $\cos 0, \cos\frac{\pi}{n}, \ldots, \cos\frac{(n-1)\pi}{n}$, and $\cos\pi$.
Chebyshev's Theorem The choice of real numers
$-1 \le x_1 < x_2 < \cdots < x_n \le 1$ that makes the value of

$$\max_{-1\le x\le 1} |(x-x_1)\cdots(x-x_n)|,$$

as small as possible is to use roots of Chebyshev's n degree polynomial, i.e.,

$$x_i = \cos\frac{(2i-1)\pi}{2n} \quad \text{for} \quad i = 1, \ldots, n.$$

And the minimum value is $\frac{1}{2^{n-1}}$.
Proof. Suppose there exists another degree n polynomial $P_n(x)$ that satisfies

$$\max_{-1\le x\le 1} |P_n(x)| < \frac{1}{2^{n-1}}.$$

Then, at $\cos 0, \cos\frac{\pi}{n}, \ldots, \cos\pi$, sign of

$$f(x) = 2^{n-1}P_n(x) - T_n(x),$$

alternates between $-$ and $+$ a total of $n+1$ times, which indicates that $f(x)$ has at least distinct roots. But $f(x)$ is of degree $n-1$, and can have at most $n-1$ roots.

Wangtao Lu
Zhejiang University

Chebyshev Theorem

Prop. 4 $T_n(1) = 1$ and $T_n(-1) = (-1)^n$.

Prop. 5 $T_n(x)$ alternates between -1 and 1 a total of $n+1$ times. This happens at $\cos 0, \cos\frac{\pi}{n}, \ldots, \cos\frac{(n-1)\pi}{n}$, and $\cos\pi$.

Chebyshev's Theorem The choice of real numers $-1 \le x_1 < x_2 < \cdots < x_n \le 1$ that makes the value of

$$\max_{-1\le x\le 1} |(x-x_1)\cdots(x-x_n)|,$$

as small as possible is to use roots of Chebyshev's n degree polynomial, i.e.,

$$x_i = \cos\frac{(2i-1)\pi}{2n} \quad \text{for} \quad i = 1, \ldots, n.$$

And the minimum value is $\frac{1}{2^{n-1}}$.

Proof. Suppose there exists another degree n polynomial $P_n(x)$ that satisfies

$$\max_{-1\le x\le 1} |P_n(x)| < \frac{1}{2^{n-1}}.$$

Then, at $\cos 0, \cos\frac{\pi}{n}, \ldots, \cos\pi$, sign of

$$f(x) = 2^{n-1}P_n(x) - T_n(x),$$

alternates between $-$ and $+$ a total of $n+1$ times, which indicates that $f(x)$ has at least n distinct roots. But $f(x)$ is of degree $n-1$, and can have at most $n-1$ roots.

Change of interval

How to choose base points in $[a, b]$ for $a < b$?
or
How to find the degree n Chebyshev polynomial defined in $[a, b]$?

Wangtao Lu
Zhejiang University

Change of interval

Wangtao Lu
Zhejiang University

How to choose base points in $[a, b]$ for $a < b$?

or

How to find the degree n Chebyshev polynomial defined in $[a, b]$?

Since $T_n(x)$ is defined in $[-1, 1]$, if we let

$$t = \frac{b+a}{2} + \frac{b-a}{2}x,$$

then $t \in (a, b)$. Thus,

$$T_n(\frac{2}{b-a}(t - \frac{b+a}{2}))$$

is the required polynomial.

$$t_n = \frac{b+a}{2} + \frac{b-a}{2}\cos\frac{(2i-1)\pi}{2n}, \quad i = 1, \ldots, n,$$

are the required base points.

Example 4. Find the four Chebyshev base points for interpolation on the interval $[0, \pi/2]$, and find the upper bound for the Chebyshev interpolation error for $f(x) = \sin x$ on the interval using MATLAB.

3.4 Cubic Splines

Splines use piecewise polynomials with certain smooth conditions for interpolating given data sets. Thus, cubic splines use piecewise polynomials of degree 3 or less for interpolation.
Why piecewise but not a single polynomial to interpolate the data sets?

1. Data sets themselves cannot be modeled by a smooth function!
2. Even it can, Runge's phenomena sometimes prevents the interpolation from being successful.

Wangtao Lu
Zhejiang University

Linear Spline

Linear Splines are piecewise polynomials of degree 1 or less for interpolation. For a set of data points $\{(x_i, y_i)\}_{i=1}^n$ with $x_1 < x_2 < \cdots < x_n$, the line spline $S(x)$ that goes through those points in order are line segments that are drawn between neighboring points.

Example 1. Find the line spline $S(x)$ that interpolates $(1, 2)$, $(2, 1)$, $(4, 4)$ and $(5, 3)$.

Wangtao Lu
Zhejiang University

Cubic Splines

Assume that we are given the n data points $(x_1, y_1), \ldots, (x_n, y_n)$, where $x_1 < x_2 < \cdots < x_n$. A **cubic spline** $S(x)$ through the data points is a set of cubic polynomials.

$$
\begin{aligned}
S_1(x) &= y_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3, \quad \text{on} \quad [x_1, x_2] \\
S_2(x) &= y_2 + b_2(x - x_2) + c_2(x - x_2)^2 + d_2(x - x_2)^3, \quad \text{on} \quad [x_2, x_3] \\
&\vdots \\
S_{n-1}(x) &= y_{n-1} + b_{n-1}(x - x_{n-1}) + c_{n-1}(x - x_{n-1})^2 + d_{n-1}(x - x_{n-1})^3, \quad \text{on} \quad [x_{n-1}, x_n]
\end{aligned}
$$

with the following properties:

P0. $S_i(x_i) = y_i$ and for $i = 1, \ldots, n-1$.

P1. $S_{i-1}(x_i) = S_i(x_i)$ for $i = 2, \ldots, n-1$.

P2. $S'_{i-1}(x_i) = S'_i(x_i)$ for $i = 2, \ldots, n-1$.

P3. $S''_{i-1}(x_i) = S''_i(x_i)$ for $i = 2, \ldots, n-1$.

For example, the following functions define a cubic spline for the data points $(1,2),(2,1),(4,4)$, and $(5,3)$:

$$
\begin{aligned}
S_1(x) &= 2 - \frac{13}{8}(x-1) + 0(x-1)^2 + \frac{5}{8}(x-1)^3, x \in [1,2],\\
S_2(x) &= 1 + \frac{1}{4}(x-2) + \frac{15}{8}(x-2)^2 - \frac{5}{8}(x-2)^3, x \in [2,4],\\
S_3(x) &= 4 + \frac{1}{4}(x-4) - \frac{15}{8}(x-4)^2 + \frac{5}{8}(x-4)^3, x \in [4,5].
\end{aligned}
$$

Let's check the Properties in MATLAB.

Linear System

How many constraints are given in the Properties?______.
How many unknowns need to solve?______.

Wangtao Lu
Zhejiang University

Linear System

How many constraints are given in the Properties?$\underline{3n-5}$.
How many unknowns need to solve?______.

Linear System

How many constraints are given in the Properties? $\underline{3n-5}$.
How many unknowns need to solve? $\underline{3n-3}$.
The linear system therefore is !
We need extra conditions!

Linear System

How many constraints are given in the Properties?$\underline{3n-5}$.
How many unknowns need to solve?$\underline{3n-3}$.
The linear system therefore is underdetermined!
We need extra conditions!

Linear System

Wangtao Lu
Zhejiang University

How many constraints are given in the Properties?$\underline{3n-5}$.
How many unknowns need to solve?$\underline{3n-3}$.
The linear system therefore is underdetermined!
We need two extra conditions!

Natural Spline Condition

We add two more conditions at the two endpoints:

$$\mathrm{P4}: S_1''(x_1) = 0, \quad S_{n-1}''(x_n) = 0.$$

We now have $3n-3$ constraints and $3n-3$ unknowns. Let's build up the linear system now.

P1 implies for $j = 1, \ldots, n-1$

$$b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3 = y_{j+1} - y_j$$

P2 implies for $j = 1, \ldots, n-2$

$$b_j + 2c_j(x_{j+1} - x_j) + 3d_j(x_{j+1} - x_j)^2 - b_{j+1} = 0.$$

P3 implies for $j = 1, \ldots, n-2$

$$2c_j + 6d_j(x_{j+1} - x_j) - 2c_{j+1} = 0.$$

Natural Spline Condition

We add two more conditions at the two endpoints:

$$\mathrm{P4}: S_1''(x_1) = 0, \quad S_{n-1}''(x_n) = 0.$$

We now have $3n-3$ constraints and $3n-3$ unknowns. Let's build up the linear system now. Denote $\delta_j = x_{j+1} - x_j$ and $\Delta_j = y_{j+1} - y_j$, $j = 1, \ldots, n-1$.

P1 implies for $j = 1, \ldots, n-1$

$$b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3 = y_{j+1} - y_j$$

P2 implies for $j = 1, \ldots, n-2$

$$b_j + 2c_j(x_{j+1} - x_j) + 3d_j(x_{j+1} - x_j)^2 - b_{j+1} = 0.$$

P3 implies for $j = 1, \ldots, n-2$

$$2c_j + 6d_j(x_{j+1} - x_j) - 2c_{j+1} = 0.$$

Wangtao Lu
Zhejiang University

Natural Spline Condition

We add two more conditions at the two endpoints:

$$\text{P4}: S_1''(x_1) = 0, \quad S_{n-1}''(x_n) = 0.$$

We now have $3n-3$ constraints and $3n-3$ unknowns. Let's build up the linear system now. Denote $\delta_j = x_{j+1} - x_j$ and $\Delta_j = y_{j+1} - y_j$, $j = 1, \dots, n-1$.

P1 implies for $j = 1, \dots, n-1$

$$b_j\delta_j + c_j\delta_j^2 + d_j\delta_j^3 = \Delta_j$$

P2 implies for $j = 1, \dots, n-2$

$$b_j + 2c_j\delta_j + 3d_j\delta_j^2 - b_{j+1} = 0$$

P3 implies for $j = 1, \dots, n-2$

$$2c_j + 6d_j\delta_j - 2c_{j+1} = 0$$

P4 implies that

$$2c_1 = 0, \quad 2c_{n-1} + 6d_{n-1}\delta_{n-1} = 0$$

Wangtao Lu
Zhejiang University

Natural Spline Condition

We add two more conditions at the two endpoints:

$$\mathrm{P4}: S_1''(x_1) = 0, \quad S_{n-1}''(x_n) = 0.$$

We now have $3n - 3$ constraints and $3n - 3$ unknowns. Let's build up the linear system now. Denote $\delta_j = x_{j+1} - x_j$ and $\Delta_j = y_{j+1} - y_j$, $j = 1, \ldots, n-1$.

P1 implies for $j = 1, \ldots, n-1$

$$b_j\delta_j + c_j\delta_j^2 + d_j\delta_j^3 = \Delta_j$$

P2 implies for $j = 1, \ldots, n-2$

$$b_j + 2c_j\delta_j + 3d_j\delta_j^2 - b_{j+1} = 0$$

P3 implies for $j = 1, \ldots, n-2$

$$2c_j + 6d_j\delta_j - 2c_{j+1} = 0 \to d_j = \frac{c_{j+1} - c_j}{3\delta_j}$$

P4 implies that

$$2c_1 = 0, \quad 2c_{n-1} + 6d_{n-1}\delta_{n-1} = 0$$

Wangtao Lu
Zhejiang University

Natural Spline Condition

We add two more conditions at the two endpoints:

$$\mathrm{P4}: S_1''(x_1) = 0, \quad S_{n-1}''(x_n) = 0.$$

We now have $3n-3$ constraints and $3n-3$ unknowns. Let's build up the linear system now. Denote $\delta_j = x_{j+1} - x_j$ and $\Delta_j = y_{j+1} - y_j$, $j = 1, \ldots, n-1$.

P1 implies for $j = 1, \ldots, n-1$

$$b_j\delta_j + c_j\delta_j^2 + d_j\delta_j^3 = \Delta_j$$

P2 implies for $j = 1, \ldots, n-2$

$$b_j + 2c_j\delta_j + 3d_j\delta_j^2 - b_{j+1} = 0$$

P3 implies for $j = 1, \ldots, n-2$

$$2c_j + 6d_j\delta_j - 2c_{j+1} = 0 \rightarrow d_j = \frac{c_{j+1} - c_j}{3\delta_j}$$

P4 implies that

$$2c_1 = 0, \quad 2c_{n-1} + 6d_{n-1}\delta_{n-1} = 0 \rightarrow d_{n-1} = \frac{0 - c_{n-1}}{3\delta_{n-1}}$$

Natural Spline Condition

We add two more conditions at the two endpoints:

$$\text{P4}: S_1''(x_1) = 0, \quad S_{n-1}''(x_n) = 0.$$

We now have $3n-3$ constraints and $3n-3$ unknowns. Let's build up the linear system now. Denote $\delta_j = x_{j+1} - x_j$ and $\Delta_j = y_{j+1} - y_j$, $j = 1, \ldots, n-1$.

P1 implies for $j = 1, \ldots, n-1$

$$b_j\delta_j + c_j\delta_j^2 + d_j\delta_j^3 = \Delta_j$$

P2 implies for $j = 1, \ldots, n-2$

$$b_j + 2c_j\delta_j + 3d_j\delta_j^2 - b_{j+1} = 0$$

P3 implies for $j = 1, \ldots, n-2$

$$2c_j + 6d_j\delta_j - 2c_{j+1} = 0 \rightarrow d_j = \frac{c_{j+1} - c_j}{3\delta_j}$$

P4 implies that (Introducing $2c_n = S_{n-1}''(x_n)$)

$$2c_1 = 0, \quad \left[2c_{n-1} + 6d_{n-1}\delta_{n-1} - 2c_n = 0 \rightarrow d_{n-1} = \frac{c_n - c_{n-1}}{3\delta_{n-1}}\right], 2c_n = 0.$$

Natural Spline Condition

We add two more conditions at the two endpoints:

$$\mathrm{P4}: S_1''(x_1) = 0, \quad S_{n-1}''(x_n) = 0.$$

We now have $3n-3$ constraints and $3n-3$ unknowns. Let's build up the linear system now. Denote $\delta_j = x_{j+1} - x_j$ and $\Delta_j = y_{j+1} - y_j$, $j = 1, \ldots, n-1$.

P1 implies for $j = 1, \ldots, n-1$

$$b_j\delta_j + c_j\delta_j^2 + d_j\delta_j^3 = \Delta_j$$

P2 implies for $j = 1, \ldots, n-2$

$$b_j + 2c_j\delta_j + 3d_j\delta_j^2 - b_{j+1} = 0$$

P3 implies for $j = 1, \ldots, n-2$

$$2c_j + 6d_j\delta_j - 2c_{j+1} = 0 \rightarrow d_j = \frac{c_{j+1} - c_j}{3\delta_j}$$

P4 implies that (Introducing $2c_n = S_{n-1}''(x_n)$)

$$2c_1 = 0, \quad \left[2c_{n-1} + 6d_{n-1}\delta_{n-1} - 2c_n = 0 \rightarrow d_{n-1} = \frac{c_n - c_{n-1}}{3\delta_{n-1}}\right], 2c_n = 0.$$

Natural Spline Condition

We add two more conditions at the two endpoints:

$$\mathrm{P4}: S_1''(x_1) = 0, \quad S_{n-1}''(x_n) = 0.$$

We now have $3n-3$ constraints and $3n-3$ unknowns. Let's build up the linear system now. Denote $\delta_j = x_{j+1} - x_j$ and $\Delta_j = y_{j+1} - y_j$, $j = 1, \ldots, n-1$.

P1 implies for $j = 1, \ldots, n-1$

$$b_j\delta_j + c_j\delta_j^2 + d_j\delta_j^3 = \Delta_j$$

P2 implies for $j = 1, \ldots, n-2$

$$b_j + 2c_j\delta_j + 3d_j\delta_j^2 - b_{j+1} = 0$$

P3 implies for $j = 1, \ldots, n-2, n-1$

$$2c_j + 6d_j\delta_j - 2c_{j+1} = 0 \rightarrow d_j = \frac{c_{j+1} - c_j}{3\delta_j}$$

P4 implies that (Introducing $2c_n = S_{n-1}''(x_n)$)

$$2c_1 = 0, 2c_n = 0.$$

Natural Spline Condition

We add two more conditions at the two endpoints:

$$\mathrm{P4}: S_1''(x_1) = 0, \quad S_{n-1}''(x_n) = 0.$$

We now have $3n-3$ constraints and $3n-3$ unknowns. Let's build up the linear system now. Denote $\delta_j = x_{j+1} - x_j$ and $\Delta_j = y_{j+1} - y_j$, $j = 1, \ldots, n-1$.

P1 implies for $j = 1, \ldots, n-1$

$$b_j\delta_j + c_j\delta_j^2 + d_j\delta_j^3 = \Delta_j \to b_j = \frac{\Delta_j}{\delta_j} - c_j\delta_j - d_j\delta_j^2 = \frac{\Delta_j}{\delta_j} - \frac{\delta_j}{3}(2c_j + c_{j+1})$$

P2 implies for $j = 1, \ldots, n-2$

$$b_j + 2c_j\delta_j + 3d_j\delta_j^2 - b_{j+1} = 0$$

P3 implies for $j = 1, \ldots, n-2, n-1$

$$2c_j + 6d_j\delta_j - 2c_{j+1} = 0 \to d_j = \frac{c_{j+1} - c_j}{3\delta_j}$$

P4 implies that (Introducing $2c_n = S_{n-1}''(x_n)$)

$$2c_1 = 0, 2c_n = 0.$$

Natural Spline Condition

We add two more conditions at the two endpoints:

$$\mathrm{P4}: S_1''(x_1) = 0, \quad S_{n-1}''(x_n) = 0.$$

We now have $3n-3$ constraints and $3n-3$ unknowns. Let's build up the linear system now. Denote $\delta_j = x_{j+1} - x_j$ and $\Delta_j = y_{j+1} - y_j$, $j = 1, \ldots, n-1$.

P1 implies for $j = 1, \ldots, n-1$

$$b_j\delta_j + c_j\delta_j^2 + d_j\delta_j^3 = \Delta_j \rightarrow b_j = \frac{\Delta_j}{\delta_j} - c_j\delta_j - d_j\delta_j^2 = \frac{\Delta_j}{\delta_j} - \frac{\delta_j}{3}(2c_j + c_{j+1})$$

P2 implies for $j = 1, \ldots, n-2$

$$b_j + 2c_j\delta_j + 3d_j\delta_j^2 - b_{j+1} = 0 \rightarrow \delta_j c_j + 2(\delta_j + \delta_{j+1})c_{j+1} + \delta_{j+2}c_{j+2} = 3\left(\frac{\Delta_{j+1}}{\delta_{j+1}} - \frac{\Delta_j}{\delta_j}\right)$$

P3 implies for $j = 1, \ldots, n-2, n-1$

$$2c_j + 6d_j\delta_j - 2c_{j+1} = 0 \rightarrow d_j = \frac{c_{j+1} - c_j}{3\delta_j}$$

P4 implies that (Introducing $2c_n = S_{n-1}''(x_n)$)

$$2c_1 = 0, 2c_n = 0.$$

Wangtao Lu
Zhejiang University

Final Linear System

In matrix form, we get

$$\begin{bmatrix} 1 & 0 & 0 & & & \\ \delta_1 & 2\delta_1+2\delta_2 & \delta_2 & \ddots & & \\ 0 & \delta_2 & 2\delta_2+2\delta_3 & \delta_3 & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & & \delta_{n-2} & 2\delta_{n-2}+2\delta_{n-1} & \delta_{n-1} \\ & & & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-1} c_n \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 3\left(\frac{\Delta_2}{\delta_2} - \frac{\Delta_1}{\delta_1}\right) \\ 3\left(\frac{\Delta_3}{\delta_3} - \frac{\Delta_2}{\delta_2}\right) \\ \vdots \\ 3\left(\frac{\Delta_{n-1}}{\delta_{n-1}} - \frac{\Delta_{n-2}}{\delta_{n-2}}\right) \\ 0 \end{bmatrix}$$

Theorem. Let $n \geq 2$. For a set of data points $(x_1, y_1), \dots, (x_n, y_n)$ with $x_1 < x_2 < \cdots < x_n$, there is a unique natural cubic spline fitting/interpolating the points.

Proof. Why?

An example

Example 3. Find the natural cubic spline through $(0, 3), (1, -2)$, and $(2, 1)$.

Endpoints Conditions

P4a. Natural cubic spline.

$$S_1''(x_1) = 0, \quad S_{n-1}''(x_n) = 0.$$

P4b. Curvature-adjusted cubic splines.

$$S_1''(x_1) = v_1, \quad S_{n-1}''(x_n) = v_n.$$

P4c. Clamped cubic spline.

$$S_1'(x_1) = v_1, \quad S_{n-1}'(x_n) = v_n.$$

P4d. Parabolically terminated cubic spline.

$$d_1 = 0 = d_{n-1}.$$

P4e. Not-a-knot cubic spline.

$$d_1 = d_2, \quad d_{n-2} = d_{n-1}.$$

Wangtao Lu
Zhejiang University

An example

Wangtao Lu
Zhejiang University

Example 4. Plot the five aforementioned types of cubic splines through $(0,3),(1,1),(2,4),(3,1),(4,2)$, and $(5,0)$ in MATLAB.

Assignment IV

Assignment IV:
P151, 3.1 Computer Problems: 3.
P157, 3.2 Computer Problems: 1.
P166, 3.3 Computer Problems: 2.
P178, 3.4 Computer Problems: 1.
Due date: October 21st, 2020.

Wangtao Lu
Zhejiang University