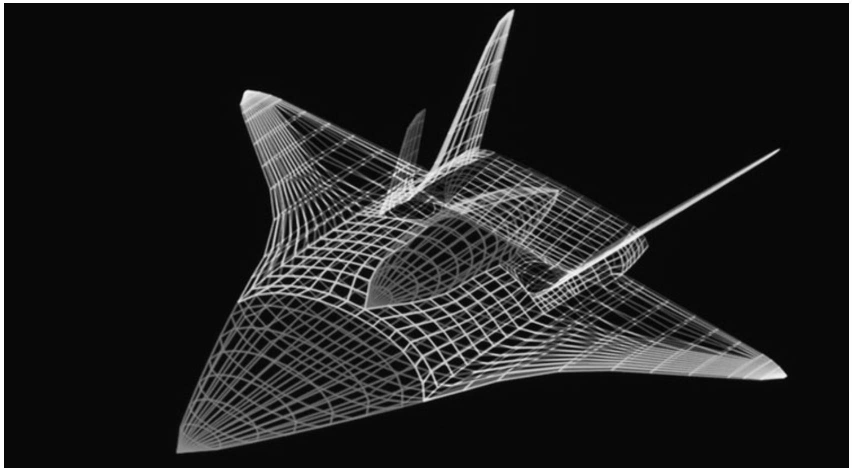
Numerical Analysis

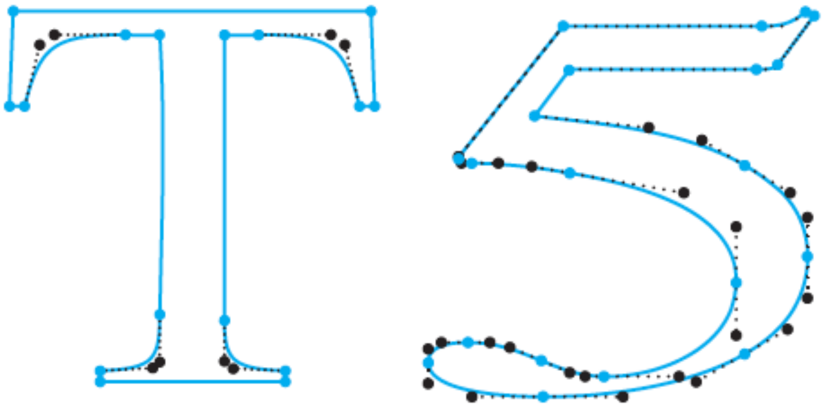
SMIE SYSU
Chang-Dong Wang
Homepage: www.scholat.com/~ChangDongWang
Course website: www.scholat.com/course/SMIENA
Email: wangchd3@mail.sysu.edu.cn
QQ Group: 342983926

Outline

- April 8
- April 15
- April 21
- April 29

Interpolation S1





- Building car, ship, aircraft
- Building fonts
- …

Interpolation S2

- The 1st question is what is the interpolation?
- Interpolation can be viewed as data compression, i.e., approximating data by a function, e.g., polynomial.
- A function is said to interpolate a set of data points if it passes through those points.
- Finding a polynomial through the set of data means replacing the information with a rule that can be evaluated in a finite number of steps.

Interpolation S3

- Data and Interpolating Functions
- Interpolation Error
- Chebyshev Interpolation
- Cubic Splines
- Bézier Curves

Data and Interpolating Functions S1

- Definition

The function $y = P(x)$ **interpolates** the data points $(x_1, y_1), \ldots, (x_n, y_n)$ if $P(x_i) = y_i$ for each $1 \le i \le n$. ❐

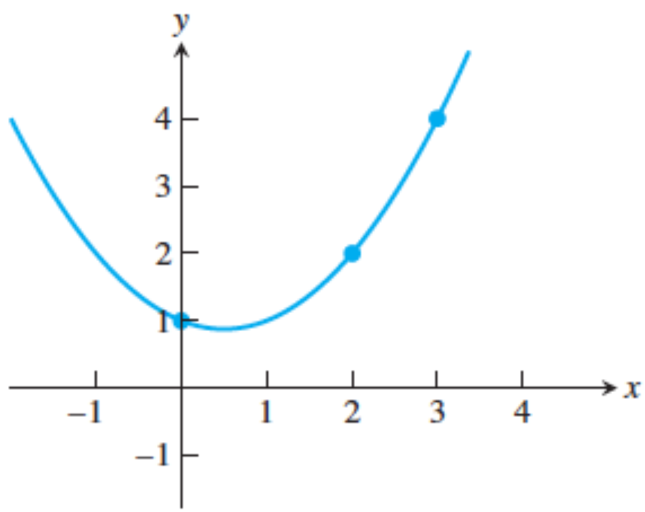


Figure 3.1 Interpolation by parabola. The points (0,1), (2,2), and (3,4) are interpolated by the function $P(x) = \frac{1}{2}x^2 - \frac{1}{2}x + 1$.

Data and Interpolating Functions S2

- Interpolating polynomial: the function is a polynomial.
- There always exists a polynomial given a set of data points with distinct x-coordinates.
- Interpolating polynomial is the reverse of polynomial evaluation.
- Why use polynomials? Straightforward mathematical properties: only adding and multiplying, which are fundamental in computer hardware.

Data and Interpolating Functions S3

- Lagrange interpolation
- Newton's divided differences
- How many degree *d* polynomials pass through *n* points?
- Code for interpolation
- Representing functions by approximating polynomials

Lagrange interpolation S1

Assume that n data points $(x_1, y_1), \ldots, (x_n, y_n)$ are given, and that we would like to find an interpolating polynomial. There is an explicit formula, called the Lagrange interpolating formula, for writing down a polynomial of degree $d = n - 1$ that interpolates the points. For example, suppose that we are given three points $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Then the polynomial

$$P_2(x) = y_1 \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + y_2 \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} + y_3 \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} \tag{3.1}$$

is the **Lagrange interpolating polynomial** for these points.

Lagrange interpolation S2

- Example

Find an interpolating polynomial for the data points (0, 1), (2, 2), and (3, 4) in Figure 3.1.

Substituting into Lagrange's formula (3.1) yields

$$\begin{aligned}
P_2(x) &= 1\frac{(x-2)(x-3)}{(0-2)(0-3)} + 2\frac{(x-0)(x-3)}{(2-0)(2-3)} + 4\frac{(x-0)(x-2)}{(3-0)(3-2)} \\
&= \frac{1}{6}(x^2-5x+6) + 2\left(-\frac{1}{2}\right)(x^2-3x) + 4\left(\frac{1}{3}\right)(x^2-2x) \\
&= \frac{1}{2}x^2 - \frac{1}{2}x + 1.
\end{aligned}$$

Check that $P_2(0) = 1$, $P_2(2) = 2$, and $P_2(3) = 4$. ◀

Lagrange interpolation S3

In general, suppose that we are presented with n points $(x_1, y_1), \ldots, (x_n, y_n)$. For each k between 1 and n, define the degree $n-1$ polynomial

$$L_k(x) = \frac{(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}.$$

The interesting property of L_k is that $L_k(x_k) = 1$, while $L_k(x_j) = 0$, where x_j is any of the other data points. Then define the degree $n-1$ polynomial

$$P_{n-1}(x) = y_1 L_1(x) + \cdots + y_n L_n(x).$$

This is a straightforward generalization of the polynomial in (3.1) and works the same way. Substituting x_k for x yields

$$P_{n-1}(x_k) = y_1 L_1(x_k) + \cdots + y_n L_n(x_k) = 0 + \cdots + 0 + y_k L_k(x_k) + 0 + \cdots + 0 = y_k,$$

so it works as designed.

Is this polynomial the only one of degree at most n-1?

Lagrange interpolation S4

- Theorem

Main Theorem of Polynomial Interpolation. 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 $i = 1, \ldots, n$. ■

- Example

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

Collinear

The Lagrange form is as follows:

$$
\begin{aligned}
P(x) &= 2\frac{(x-1)(x-2)(x-3)}{(0-1)(0-2)(0-3)} + 1\frac{(x-0)(x-2)(x-3)}{(1-0)(1-2)(1-3)} \\
&\quad + 0\frac{(x-0)(x-1)(x-3)}{(2-0)(2-1)(2-3)} - 1\frac{(x-0)(x-1)(x-2)}{(3-0)(3-1)(3-2)} \\
&= -\frac{1}{3}(x^3 - 6x^2 + 11x - 6) + \frac{1}{2}(x^3 - 5x^2 + 6x) - \frac{1}{6}(x^3 - 3x^2 + 2x) \\
&= -x + 2.
\end{aligned}
$$

Newton's divided differences S1

$$L_k(x) = \frac{(x - x_1)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n)}{(x_k - x_1)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_n)}.$$

$$P_{n-1}(x) = y_1 L_1(x) + \cdots + y_n L_n(x).$$

- Shortcomings

1. Huge computational complexity.
2. Cannot update the polynomial efficiently when adding new data points.

Newton's divided differences S2

- Definition

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

- In the previous example:

interpolating polynomial for the data points (0, 1), (2, 2), and (3, 4)

$$P_2(x) = \frac{1}{2}x^2 - \frac{1}{2}x + 1 \qquad f[0\ 2\ 3] = 1/2$$

$f[0\ 3\ 2]=f[0\ 2\ 3]=f[2\ 0\ 3]=f[2\ 3\ 0]=f[3\ 0\ 2]=f[3\ 2\ 0]=1/2$

Newton's divided differences S3

Newton's divided difference formula

$$
\begin{aligned}
P(x) = f[x_1] &+ f[x_1\ \ x_2](x - x_1) \\
&+ f[x_1\ \ x_2\ \ x_3](x - x_1)(x - x_2) \\
&+ f[x_1\ \ x_2\ \ x_3\ \ x_4](x - x_1)(x - x_2)(x - x_3) \\
&+ \cdots \\
&+ f[x_1 \cdots x_n](x - x_1)\cdots(x - x_{n-1}).
\end{aligned} \tag{3.2}
$$

- How to calculate the coefficient $f[x_1\ x_2\ \ldots\ x_n]$? You may try to substitute the data points into

(3.2) and see how to find the coefficient…

Newton's divided differences S4

the coefficients $f[x_1 \ldots x_k]$ from the above definition can be recursively calculated as follows. List the data points in a table:

$$\begin{array}{c|c} x_1 & f(x_1) \\ x_2 & f(x_2) \\ \vdots & \vdots \\ x_n & f(x_n). \end{array}$$

Now define the divided differences, which are the real numbers

$$\begin{aligned} f[x_k] &= f(x_k) \\ f[x_k \ x_{k+1}] &= \frac{f[x_{k+1}] - f[x_k]}{x_{k+1} - x_k} \\ f[x_k \ x_{k+1} \ x_{k+2}] &= \frac{f[x_{k+1} \ x_{k+2}] - f[x_k \ x_{k+1}]}{x_{k+2} - x_k} \\ f[x_k \ x_{k+1} \ x_{k+2} \ x_{k+3}] &= \frac{f[x_{k+1} \ x_{k+2} \ x_{k+3}] - f[x_k \ x_{k+1} \ x_{k+2}]}{x_{k+3} - x_k}, \end{aligned} \tag{3.3}$$

and so on.

Newton's divided differences S5

Newton's divided differences

Given $x = [x_1, \ldots, x_n]$, $y = [y_1, \ldots, y_n]$

for $j = 1, \ldots, n$
  $f[x_j] = y_j$
end

for $i = 2, \ldots, n$
  for $j = 1, \ldots, n + 1 - i$
    $f[x_j \ldots x_{j+i-1}] = (f[x_{j+1} \ldots x_{j+i-1}] - f[x_j \ldots x_{j+i-2}])/(x_{j+i-1} - x_j)$
  end
end
The interpolating polynomial is

$$P(x) = \sum_{i=1}^{n} f[x_1 \underline{\ldots\ x_i}](x - x_1) \cdots (x - \underline{x_{i-1}})$$

Newton’s divided differences S6

The recursive definition of the Newton’s divided differences allows arrangement into a convenient table. For three points the table has the form

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

The coefficients of the polynomial (3.2) can be read from the top edge of the triangle.

Newton's divided differences S7

- Example

Use divided differences to find the interpolating polynomial passing through the points (0, 1), (2, 2), (3, 4).

Applying the definitions of divided differences leads to the following table:

$$\frac{2-1}{2-0} = \frac{1}{2}$$

$$\frac{2-\frac{1}{2}}{3-0} = \frac{1}{2}$$

$$\frac{4-2}{3-2} = 2.$$

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

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

Newton's divided differences S8

- Example

Add the fourth data point $(1, 0)$ to the list in Example 3.3.

We can keep the calculations that were already done and just add a new bottom row to the triangle:

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

$$P_3(x) = P_2(x) - \frac{1}{2}(x-0)(x-2)(x-3)$$

How many degree *d* polynomials pass through *n* points?

- For *n* points, there is one and only one interpolating polynomial of degree *n*-1 or less, according to Main Theorem of Polynomial Interpolation.
- For *n* points, there are infinitely many interpolating polynomials of degree *n* or greater: Just add the extract polynomial of the desired degree that passes the *n* points to the existing polynomial.

Any degree 3 polynomial of the form $P_3(x) = P_2(x) + cx(x-2)(x-3)$ with $c \neq 0$ will pass through $(0, 1)$, $(2, 2)$, and $(3, 4)$.

Code for Interpolation

- For computing the coefficients: newtdd.m
- We use the data points from Example 3.3 to test this code: testnewtdd.m
- The program clickinterp.m uses Matlab 's graphics capability to plot the interpolation polynomial as it is being created.

Representing functions by approximating polynomials S1

- Example

Interpolate the function $f(x) = \sin x$ at 4 equally spaced points on $[0, \pi/2]$.

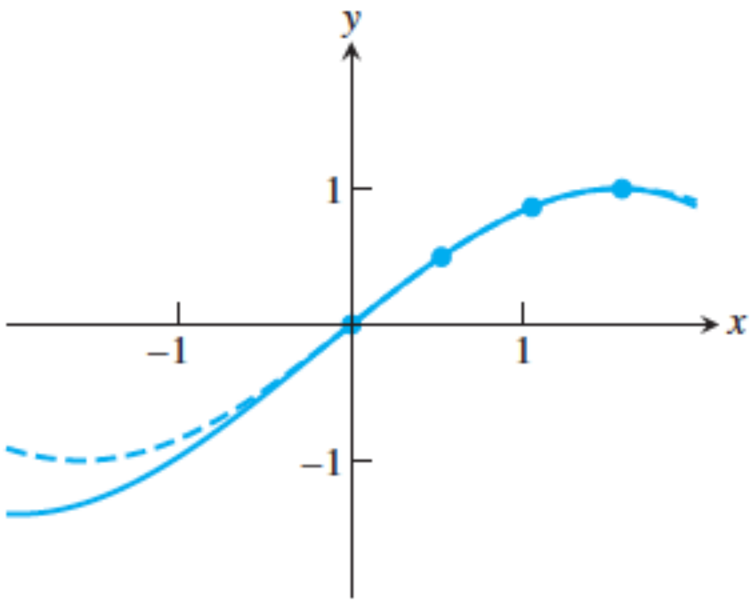
Let's compress the sine function on the interval $[0, \pi/2]$. Take four data points at equally spaced points and form the divided difference triangle. We list the values to four correct places:

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

The degree 3 interpolating polynomial is therefore

$$
\begin{aligned}
P_3(x) &= 0 + 0.9549x - 0.2443x(x - \pi/6) - 0.1139x(x - \pi/6)(x - \pi/3) \\
&= 0 + x(0.9549 + (x - \pi/6)(-0.2443 + (x - \pi/3)(-0.1139))). \qquad (3.5)
\end{aligned}
$$

Representing functions by approximating polynomials S2



x	$\sin x$	`sin1(x)`	error
1	0.8415	0.8411	0.0004
2	0.9093	0.9102	0.0009
3	0.1411	0.1428	0.0017
4	–0.7568	–0.7557	0.0011
14	0.9906	0.9928	0.0022
1000	0.8269	0.8263	0.0006

$[0, \pi/2]$ $\quad \sin x$

$[\pi/2, \pi]$ $\quad \sin x = \sin(\pi - x)$

$[\pi, 2\pi]$ $\quad \sin x = -\sin(2\pi - x)$

Coding Assignment: Computer problem 1, 3 @ page 151.

Periodic function: repeat its evaluation.

Interpolation Error

- Recall

x	$\sin x$	`sin1(x)`	error
1	0.8415	0.8411	0.0004
2	0.9093	0.9102	0.0009
3	0.1411	0.1428	0.0017
4	–0.7568	–0.7557	0.0011
14	0.9906	0.9928	0.0022
1000	0.8269	0.8263	0.0006

1. Interpolation error formula
2. Runge phenomenon

Interpolation error formula S1

Assume that we start with a function $y = f(x)$ and take data points from it to build an interpolating polynomial $P(x)$. The **interpolation error** at x is $f(x) - P(x)$.

- Theorem

Assume that $P(x)$ is the (degree $n-1$ or less) interpolating polynomial fitting the n points $(x_1, y_1), \ldots, (x_n, y_n)$. The interpolation error is

$$f(x) - P(x) = \frac{(x - x_1)(x - x_2)\cdots(x - x_n)}{n!} f^{(n)}(c), \tag{3.6}$$

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

Interpolation error formula S2

- Get back to the previous example

$$\sin x - P(x) = \frac{(x-0)\left(x-\frac{\pi}{6}\right)\left(x-\frac{\pi}{3}\right)\left(x-\frac{\pi}{2}\right)}{4!} f''''(c) \qquad \text{where } 0 < c < \pi/2.$$

The upper bound on interpolation error:

$$|\sin x - P(x)| \le \frac{\left|(x-0)\left(x-\frac{\pi}{6}\right)\left(x-\frac{\pi}{3}\right)\left(x-\frac{\pi}{2}\right)\right|}{24}|1|.$$

$f''''(c) = \sin c$

$|\sin c|$ is no more than 1

$$\text{At } x = 1 \quad |\sin 1 - P(1)| \le \frac{\left|(1-0)\left(1-\frac{\pi}{6}\right)\left(1-\frac{\pi}{3}\right)\left(1-\frac{\pi}{2}\right)\right|}{24}|1| \approx 0.0005348.$$

The actual error at $x = 1$ was .0004.

Interpolation error formula S3

We expect smaller errors when x is closer to the middle of the interval of x_i's than when it is near one of the ends because there will be more small terms in the product.

For example, we compare

$$\boxed{|\sin 0.2 - P(0.2)| \le \frac{\left|(.2-0)\left(.2-\frac{\pi}{6}\right)\left(.2-\frac{\pi}{3}\right)\left(.2-\frac{\pi}{2}\right)\right|}{24}|1| \approx 0.00313}$$

$$|\sin 0.2 - P(0.2)| = |0.19867 - 0.20056| = 0.00189.$$

$$\boxed{|\sin 1 - P(1)| \le \frac{\left|(1-0)\left(1-\frac{\pi}{6}\right)\left(1-\frac{\pi}{3}\right)\left(1-\frac{\pi}{2}\right)\right|}{24}|1| \approx 0.0005348.}$$

The actual error at $x = 1$ was .0004

Interpolation error formula S4

- Example

Find an upper bound for the difference at $x = 0.25$ and $x = 0.75$ between $f(x) = e^x$ and the polynomial that interpolates it at the points $-1, -0.5, 0, 0.5, 1$.

The interpolation error formula (3.6) gives

$$f(x) - P_4(x) = \frac{(x+1)\left(x+\frac{1}{2}\right)x\left(x-\frac{1}{2}\right)(x-1)}{5!} f^{(5)}(c),$$

where $-1 < c < 1$. The fifth derivative is $f^{(5)}(c) = e^c$. Since e^x is increasing with x, its maximum is at the right-hand end of the interval, so $|f^{(5)}| \le e^1$ on $[-1, 1]$. For $-1 \le x \le 1$, the error formula becomes

$$|e^x - P_4(x)| \le \frac{(x+1)\left(x+\frac{1}{2}\right)x\left(x-\frac{1}{2}\right)(x-1)}{5!} e.$$

Interpolation error formula S5

- Example (cont…)

At $x = 0.25$, the interpolation error has the upper bound

$$
\begin{aligned}
|e^{0.25} - P_4(0.25)| &\leq \frac{(1.25)(0.75)(0.25)(-0.25)(-0.75)}{120} e \\
&\approx .000995.
\end{aligned}
$$

At $x = 0.75$, the interpolation error is potentially larger:

$$
\begin{aligned}
|e^{0.75} - P_4(0.75)| &\leq \frac{(1.75)(1.25)(0.75)(0.25)(0.25)}{120} e \\
&\approx .002323.
\end{aligned}
$$

Note again that the interpolation error will tend to be smaller close to the center of the interpolation interval. ◀

Runge phenomenon S1

- Polynomial interpolation performs better in some shapes than in the other shapes.
- Run clickinterp.m with data points that cause the function to be zero at equally spaced points $x = -3, -2.5, -2, -1.5, \ldots, 2.5, 3$, except for $x = 0$, where we set a value of 1.

The polynomial that goes through points situated like this refuses to stay between 0 and 1, unlike the data points. This is an illustration of the so-called **Runge phenomenon**.

Runge phenomenon S2

- Runge Example:

Interpolate $f(x) = 1/(1 + 12x^2)$ at evenly spaced points in $[-1, 1]$.

- RungeExample.m

characteristic of the Runge phenomenon: polynomial wiggle near the ends of the interpolation interval.

Coding Assignment: Computer problem 1 @ page 157.

Chebyshev Interpolation

- Recall that in the previous examples, we choose evenly spaced base points.
- However, the choice of base point spacing has a significant effect on the interpolation error (see Eq. (3.6)). $$f(x) - P(x) = \frac{(x - x_1)(x - x_2)\cdots(x - x_n)}{n!} f^{(n)}(c)$$
- Chebyshev interpolation refers to a particular optimal way of spacing the base points so as to reduce the interpolation error.

[RungeExampleVSChebyShev.m]

Chebyshev's theorem S1

The interpolation error

$$\frac{\boxed{(x-x_1)(x-x_2)\cdots(x-x_n)}}{n!}f^{(n)}(c)$$

on the interpolation interval. Let's fix the interval to be $[-1, 1]$ for now.

Only the numerator can be changed.

The minimax problem of interpolation

to find particular $x_1, \ldots, x_n$ in $[-1, 1]$ that cause the maximum value of

$$(x-x_1)(x-x_2)\cdots(x-x_n)$$

to be as small as possible

Chebyshev's theorem S2

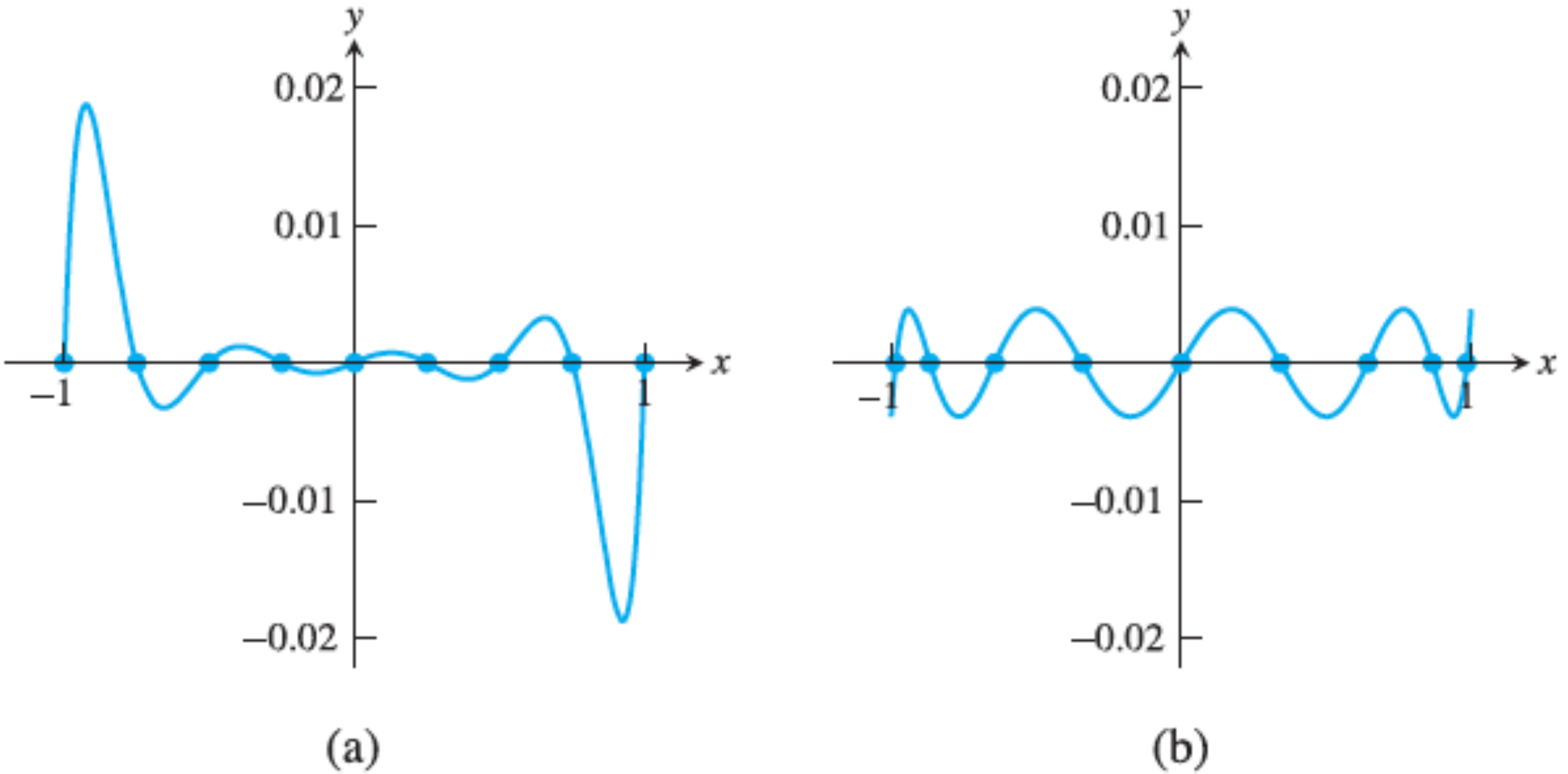


Figure 3.7 Part of the Interpolation Error Formula. Plots of $(x-x_1)\cdots(x-x_9)$ for (a) nine evenly spaced base points x_i (b) nine Chebyshev roots x_i.

Chebyshev's theorem S3

- Theorem

The choice of real numbers $-1 \le x_1, \ldots, x_n \le 1$ that makes the value of

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

as small as possible is

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

and the minimum value is $\boxed{1/2^{n-1}}$. In fact, the minimum is achieved by

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

where $T_n(x)$ denotes the degree n Chebyshev polynomial.

Chebyshev roots

$$\boxed{x_i = \cos \frac{\text{odd } \pi}{2n}}$$

We will call the interpolating polynomial that uses the Chebyshev roots as base points the **Chebyshev interpolating polynomial.**

Chebyshev's theorem S4

- Example

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

The interpolation error formula (3.6) gives

$$f(x) - P_4(x) = \frac{(x - x_1)(x - x_2)(x - x_3)(x - x_4)(x - x_5)}{5!} f^{(5)}(c),$$

where

$$x_1 = \cos\frac{\pi}{10}, \quad x_2 = \cos\frac{3\pi}{10}, \quad x_3 = \cos\frac{5\pi}{10}, \quad x_4 = \cos\frac{7\pi}{10}, \quad x_5 = \cos\frac{9\pi}{10}$$

are the Chebyshev roots and where $-1 < c < 1$. According to the Chebyshev Theorem 3.6, for $-1 \le x \le 1$,

$$|(x - x_1)\cdots(x - x_5)| \le \frac{1}{2^4}.$$

In addition, $|f^{(5)}| \le e^1$ on $[-1, 1]$. The interpolation error is

$$|e^x - P_4(x)| \le \frac{e}{2^4 5!} \approx 0.00142$$

for all x in the interval $[-1, 1]$.

At $x = 0.25$

$\approx .000995.$

At $x = 0.75$

$\approx .002323.$

Chebyshev's theorem S5

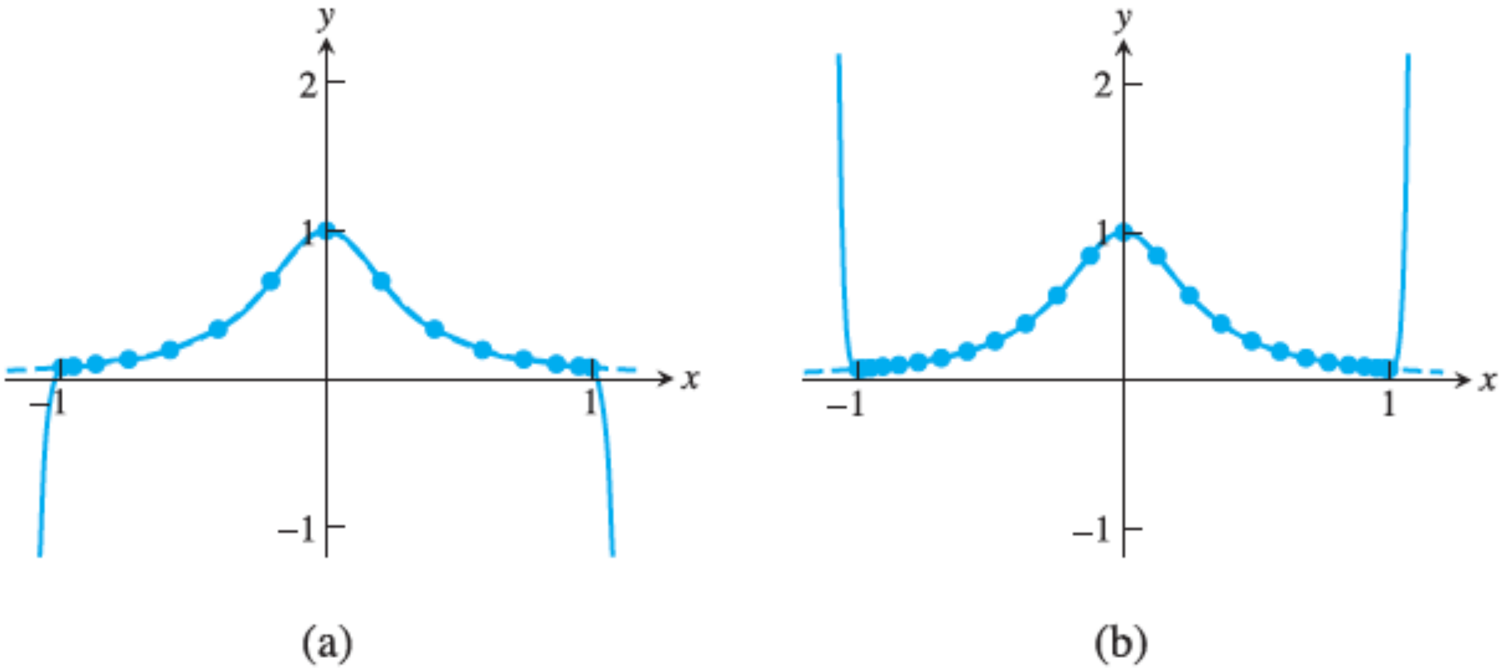


Figure 3.8 Interpolation of Runge Example with Chebyshev nodes. The Runge function $f(x) = 1/(1 + 12x^2)$ is graphed along with its Chebyshev interpolation polynomial for (a) 15 points (b) 25 points. The error on [−1, 1] is negligible at this resolution. The polynomial wiggle of Figure 3.6 has vanished, at least between −1 and 1.

RungeExampleVSChebyShev.m

Chebyshev polynomials S1

Define the nth **Chebyshev polynomial** by $T_n(x) = \cos(n \arccos x)$.

for $n = 0$ it gives the degree 0 polynomial 1

for $n = 1$ we get $T_1(x) = \cos(\arccos x) = x$.

For $n = 2$ $T_2(x) = \cos 2y = \cos^2 y - \sin^2 y = 2\cos^2 y - 1 = 2x^2 - 1$

$\boxed{\cos(a + b) = \cos a \cos b - \sin a \sin b. \text{ Set } y = \arccos x, \text{ so that } \cos y = x.}$

In general, note that

$$T_{n+1}(x) = \cos(n + 1)y = \cos(ny + y) = \cos ny \cos y - \sin ny \sin y$$
$$T_{n-1}(x) = \cos(n - 1)y = \cos(ny - y) = \cos ny \cos y - \sin ny \sin(-y).$$

Because $\sin(-y) = -\sin y$, we can add the preceding equations to get

$$T_{n+1}(x) + T_{n-1}(x) = 2\cos ny \cos y = 2xT_n(x).$$

$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$, **recursion relation** for the Chebyshev polynomials.

Chebyshev polynomials S2

- Fact 1

The T_n's are polynomials. We showed this explicitly for T_0, T_1, and T_2. Since T_3 is a polynomial combination of T_1 and T_2, T_3 is also a polynomial. The same argument goes for all T_n. The first few Chebyshev polynomials (see Figure 3.9) are

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

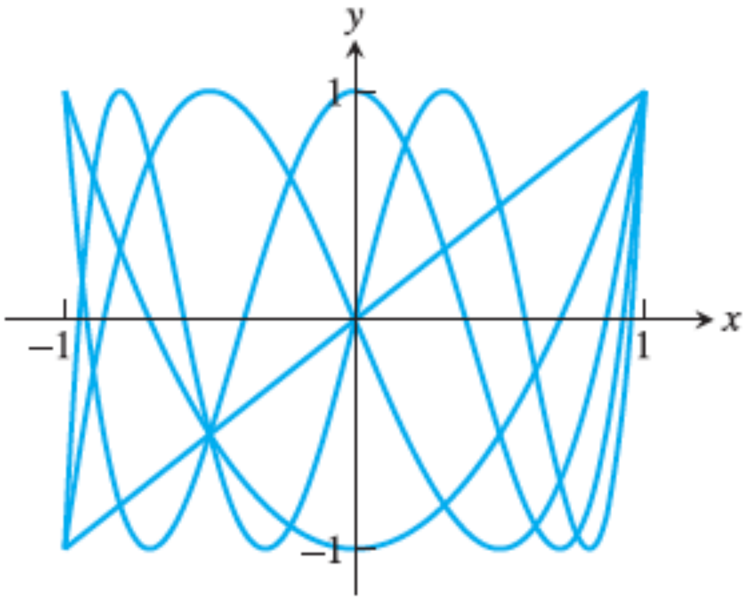


Figure 3.9 Plot of the Degree 1 through 5 Chebyshev Polynomials. Note that $T_n(1) = 1$ and the maximum absolute value taken on by $T_n(x)$ inside $[-1, 1]$ is 1.

Chebyshev polynomials S3

- Fact 2

$\deg(T_n) = n$, and the leading coefficient is 2^{n-1}. This is clear for $n = 1$ and 2, and the recursion relation extends the fact to all n. ❐

$$\boxed{T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)}$$

- Fact 3

$T_n(1) = 1$ and $T_n(-1) = (-1)^n$. Both are clear for $n = 1$ and 2. In general,

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

and

$$\begin{aligned} T_{n+1}(-1) &= 2(-1)T_n(-1) - T_{n-1}(-1) \\ &= -2(-1)^n - (-1)^{n-1} \\ &= (-1)^{n-1}(2-1) = (-1)^{n-1} = (-1)^{n+1}. \end{aligned}$$ ❐

- Fact 4

The maximum absolute value of $T_n(x)$ for $-1 \le x \le 1$ is 1. This follows immediately from the fact that $T_n(x) = \cos y$ for some y. ❐

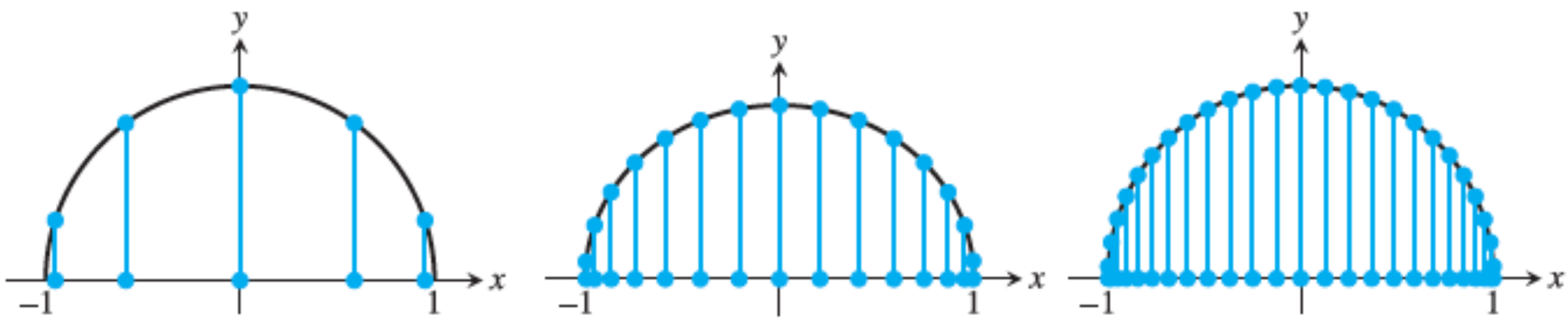
Chebyshev polynomials S4

- Fact 5

All zeros of $T_n(x)$ are located between -1 and 1. See Figure 3.10. In fact, the zeros are the solution of $0 = \cos(n \arccos x)$. Since $\cos y = 0$ if and only if $y = \text{odd integer} \cdot (\pi/2)$, we find that

$$n \arccos x = \text{odd} \cdot \pi/2$$

$$x = \cos \frac{\text{odd} \cdot \pi}{2n}.$$ Chebyshev roots



(a) (b) (c)

Figure 3.10 Location of Zeros of the Chebyshev Polynomial. The roots are the *x*-coordinates of evenly spaced points around the circle (a) degree 5 (b) degree 15 (c) degree 25.

Chebyshev polynomials S5

- Fact 6

$T_n(x)$ alternates between -1 and 1 a total of $n+1$ times. In fact, this happens at $\cos 0, \cos \pi/n, \ldots, \cos(n-1)\pi/n, \cos \pi$. ❐

It follows from Fact 2 that the polynomial $T_n(x)/2^{n-1}$ is monic (has leading coefficient 1). Since, according to Fact 5, all roots of $T_n(x)$ are real, we can write $T_n(x)/2^{n-1}$ in

factored form as $(x - x_1)\cdots(x - x_n)$ where the x_i are the Chebyshev nodes as described in Theorem 3.8.

Chebyshev's theorem follows directly from these facts.

Proof of Theorem 3.6. Let $P_n(x)$ be a monic polynomial with an even smaller absolute maximum on $[-1, 1]$; in other words, $|P_n(x)| < 1/2^{n-1}$ for $-1 \leq x \leq 1$. This assumption leads to a contradiction. Since $T_n(x)$ alternates between -1 and 1 a total of $n+1$ times (Fact 6), at these $n+1$ points the difference $P_n - T_n/2^{n-1}$ is alternately positive and negative. Therefore, $P_n - T_n/2^{n-1}$ must cross zero at least n times; that is, it must have at least n roots. This contradicts the fact that, because P_n and $T_n/2^{n-1}$ are monic, their difference is of degree $\leq n-1$.

Change of interval S1

- Basic idea: The base points are moved so that they have the same relative positions in [a, b] as that they had in [−1, 1] .
- Two main steps:

1. Stretching: Stretch the points by the factor (b − a)/2 (the ratio of the two interval lengths).
2. Translation: Translate the points by (b + a)/2 to move the center of mass from 0 to the midpoint of [a, b].

Change of interval S2

In other words, move from the original points $\cos \frac{\text{odd } \pi}{2n}$

to $\frac{b-a}{2} \cos \frac{\text{odd } \pi}{2n} + \frac{b+a}{2}$.

With the new Chebyshev base points $x_1, \ldots, x_n$ in $[a, b]$, the corresponding upper bound on the numerator of the interpolation error formula is changed due to the stretch by $(b-a)/2$ on each factor $x - x_i$. As a result, the minimax value $1/2^{n-1}$ must be replaced by $[(b-a)/2]^n/2^{n-1}$.

Chebyshev interpolation nodes

On the interval [a,b],

$$x_i = \frac{b+a}{2} + \frac{b-a}{2} \cos \frac{(2i-1)\pi}{2n}$$

for $i = 1, \ldots, n$. The inequality

$$|(x - x_1) \cdots (x - x_n)| \leq \frac{\left(\frac{b-a}{2}\right)^n}{2^{n-1}} \tag{3.14}$$

holds on $[a, b]$.

Change of interval S3

- Example

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

This is a second attempt. We used evenly spaced base points in Example 3.7. The Chebyshev base points are

$$\frac{\frac{\pi}{2} - 0}{2} \cos\left(\frac{\text{odd}\, \pi}{2(4)}\right) + \frac{\frac{\pi}{2} + 0}{2},$$

or

$$x_1 = \frac{\pi}{4} + \frac{\pi}{4}\cos\frac{\pi}{8}, x_2 = \frac{\pi}{4} + \frac{\pi}{4}\cos\frac{3\pi}{8}, x_3 = \frac{\pi}{4} + \frac{\pi}{4}\cos\frac{5\pi}{8}, x_4 = \frac{\pi}{4} + \frac{\pi}{4}\cos\frac{7\pi}{8}.$$

From (3.14), the worst-case interpolation error for $0 \le x \le \pi/2$ is

$$|\sin x - P_3(x)| = \frac{|(x - x_1)(x - x_2)(x - x_3)(x - x_4)|}{4!} |f''''(c)|$$

$$\le \frac{\left(\frac{\frac{\pi}{2} - 0}{2}\right)^4}{4! 2^3} 1 \approx 0.00198.$$

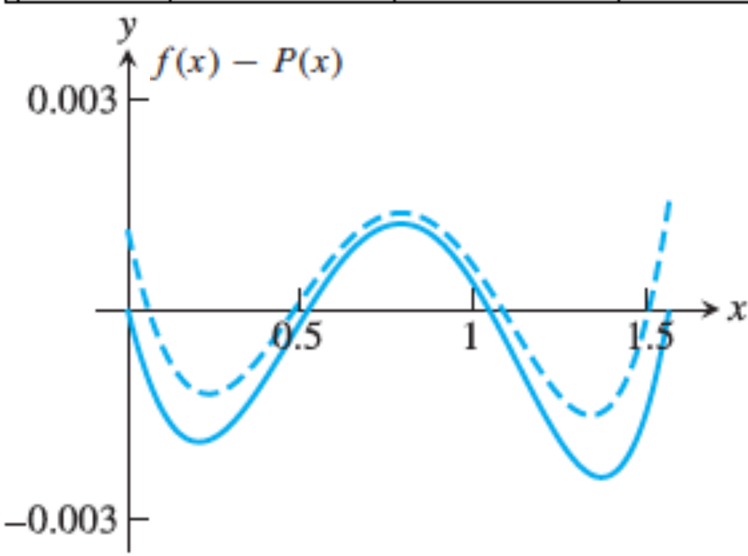
Change of interval S4

Chebyshev roots

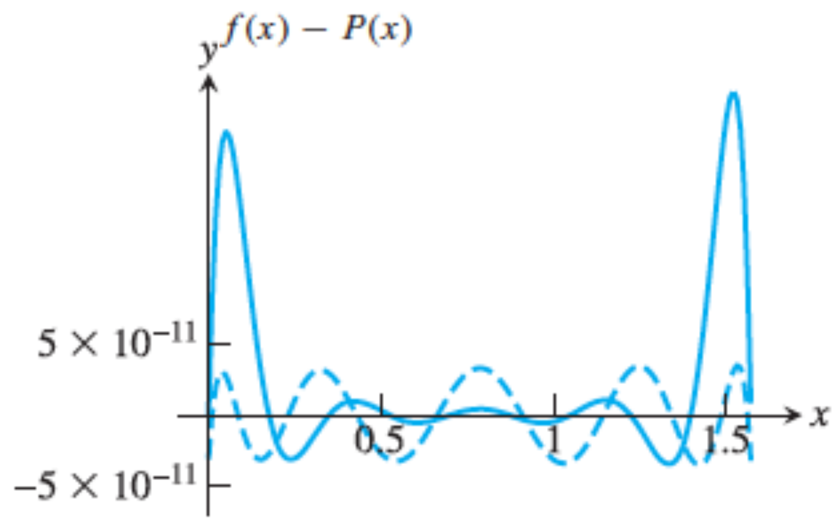
x	$\sin x$	$P_3(x)$	error
1	0.8415	0.8408	0.0007
2	0.9093	0.9097	0.0004
3	0.1411	0.1420	0.0009
4	−0.7568	−0.7555	0.0013
14	0.9906	0.9917	0.0011
1000	0.8269	0.8261	0.0008

Evenly spaced base points

x	$\sin x$	`sin1(x)`	error
1	0.8415	0.8411	0.0004
2	0.9093	0.9102	0.0009
3	0.1411	0.1428	0.0017
4	−0.7568	−0.7557	0.0011
14	0.9906	0.9928	0.0022
1000	0.8269	0.8263	0.0006



(a)



(b)

Change of interval S5

- Example

Design a sine key that will give output correct to 10 decimal places.

The goal is find n.

The maximum interpolation error for the polynomial $P_{n-1}(x)$ on the interval $[0, \pi/2]$ is

$$|\sin x - P_{n-1}(x)| = \frac{|(x-x_1)\cdots(x-x_n)|}{n!}|f^{(n)}(c)|$$
$$\leq \frac{\left(\frac{\frac{\pi}{2}-0}{2}\right)^n}{n!2^{n-1}}1.$$

for $n = 9$ the error bound is $\approx 0.1224 \times 10^{-8}$

for $n = 10$ it is $\approx 0.4807 \times 10^{-10}$

trial and error

sin2.m plotsin2.m

Writing assignment: Exercise 1 @165
Coding assignment: Computer problem 1 @165.

Cubic Splines S1

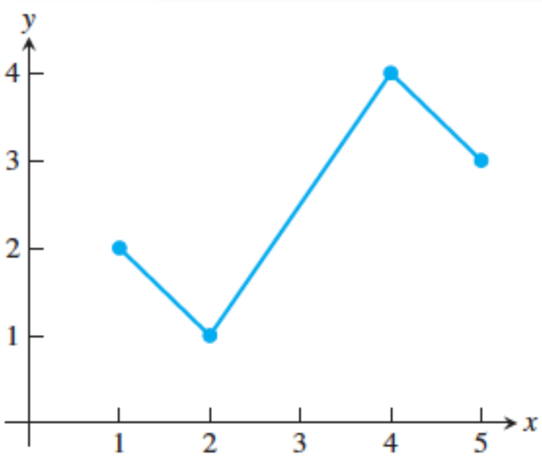
- Polynomial interpolation uses a single formula, given by a polynomial, to meet all data points.
- Spline interpolation uses several formulas, each a low-degree polynomial, to pass through the data points.

Cubic Splines S2

- Linear spline: "connects the data points" with straight-line segments.

Assume that we are given a set of data points $(x_1, y_1), \ldots, (x_n, y_n)$ with $x_1 < \cdots < x_n$. A linear spline consists of the $n - 1$ line segments that are drawn between neighboring pairs of points. $(x_i, y_i), (x_{i+1}, y_{i+1})$

$y = a_i + b_i x$ is drawn through the two points.

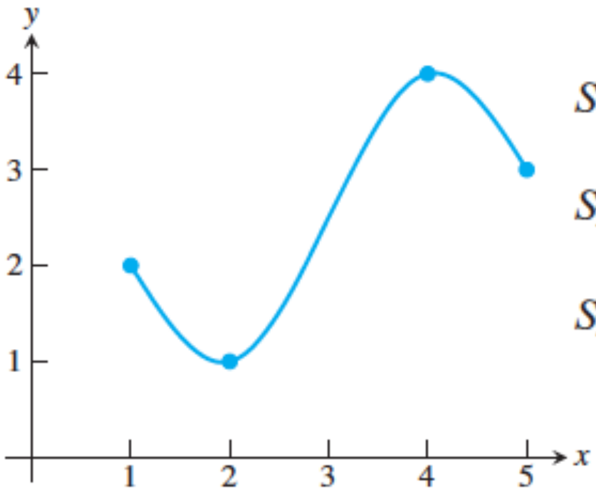


$(1, 2), (2, 1), (4, 4)$ and $(5, 3)$

$$S_1(x) = 2 - (x - 1) \text{ on } [1, 2]$$
$$S_2(x) = 1 + \frac{3}{2}(x - 2) \text{ on } [2, 4]$$
$$S_3(x) = 4 - (x - 4) \text{ on } [4, 5].$$

Cubic Splines S3

- Cubic spline: replaces linear functions between the data points by degree 3 (cubic) polynomials, so as to get smoothness.



$(1, 2), (2, 1), (4, 4)$ and $(5, 3)$

$$
\begin{aligned}
S_1(x) &= 2 - \frac{13}{8}(x-1) + 0(x-1)^2 + \frac{5}{8}(x-1)^3 \text{ on } [1, 2] \\
S_2(x) &= 1 + \frac{1}{4}(x-2) + \frac{15}{8}(x-2)^2 - \frac{5}{8}(x-2)^3 \text{ on } [2, 4] \qquad (3.16) \\
S_3(x) &= 4 + \frac{1}{4}(x-4) - \frac{15}{8}(x-4)^2 + \frac{5}{8}(x-4)^3 \text{ on } [4, 5].
\end{aligned}
$$

The smoothness is achieved by arranging for the neighboring pieces $\mathbf{S}_i$ and $\mathbf{S}_{i+1}$ of the spline to have the same zeroth, first, and second derivatives when evaluated at each knot.

Properties of splines S1

To be a little more precise about the properties of a cubic spline, we make the following definition: Assume that we are given the n data points $(x_1, y_1), \ldots, (x_n, y_n)$, where the x_i are distinct and in increasing order. A **cubic spline** $S(x)$ through the data points $(x_1, y_1), \ldots, (x_n, y_n)$ is a set of cubic polynomials

$$
\begin{aligned}
S_1(x) &= y_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3 \text{ on } [x_1, x_2] \\
S_2(x) &= y_2 + b_2(x - x_2) + c_2(x - x_2)^2 + d_2(x - x_2)^3 \text{ on } [x_2, x_3] \\
&\vdots \\
S_{n-1}(x) &= y_{n-1} + b_{n-1}(x - x_{n-1}) + c_{n-1}(x - x_{n-1})^2 + d_{n-1}(x - x_{n-1})^3 \text{ on } [x_{n-1}, x_n]
\end{aligned}
\tag{3.17}
$$

with the following properties:

Property 1 $S_i(x_i) = y_i$ and $S_i(x_{i+1}) = y_{i+1}$ for $i = 1, \ldots, n-1$.

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

Property 3 $S''_{i-1}(x_i) = S''_i(x_i)$ for $i = 2, \ldots, n-1$.

Properties of splines S2

- Example

Check that $\{S_1, S_2, S_3\}$ in (3.16) satisfies all cubic spline properties for the data points (1, 2), (2, 1), (4, 4), and (5, 3).

Property 1. There are $n = 4$ data points. We must check

$$S_1(1) = 2 \text{ and } S_1(2) = 1$$
$$S_2(2) = 1 \text{ and } S_2(4) = 4$$
$$S_3(4) = 4 \text{ and } S_3(5) = 3.$$

These follow easily from the defining equations (3.16).

Property 2. The first derivatives of the spline functions are

$$S_1'(x) = -\frac{13}{8} + \frac{15}{8}(x-1)^2$$
$$S_2'(x) = \frac{1}{4} + \frac{15}{4}(x-2) - \frac{15}{8}(x-2)^2$$
$$S_3'(x) = \frac{1}{4} - \frac{15}{4}(x-4) + \frac{15}{8}(x-4)^2.$$

Properties of splines S3

- Example (cont…)

We must check $S_1'(2) = S_2'(2)$ and $S_2'(4) = S_3'(4)$.

The first is $-\frac{13}{8} + \frac{15}{8} = \frac{1}{4}$ and the second is $\frac{1}{4} + \frac{15}{4}(4-2) - \frac{15}{8}(4-2)^2 = \frac{1}{4}$

Property 3. The second derivatives are

$$
\begin{aligned}
S_1''(x) &= \frac{15}{4}(x-1) \\
S_2''(x) &= \frac{15}{4} - \frac{15}{4}(x-2) \\
S_3''(x) &= -\frac{15}{4} + \frac{15}{4}(x-4).
\end{aligned}
$$

We must check $S_1''(2) = S_2''(2)$ and $S_2''(4) = S_3''(4)$, both of which are true. Therefore, (3.16) is a cubic spline.

Properties of splines S4

- Let's get back to the three properties:

Property 1 $S_i(x_i) = y_i$ and $\boxed{S_i(x_{i+1}) = y_{i+1}}$ for $i = 1, \ldots, n-1$.

n - 1 conditions

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

n - 2 conditions

Property 3 $\boxed{S''_{i-1}(x_i) = S''_i(x_i)}$ for $i = 2, \ldots, n-1$.

n - 2 conditions

A total of n − 1 + 2(n − 2) = 3n − 5 independent equations must be satisfied. The n – 1 formulas S_i consists of three coefficients b_i, c_i, d_i, So there are a total of 3(n – 1) = 3n – 3 variables.
Therefore, solving for the coefficients is a problem of solving 3n − 5 linear equations in 3n − 3 unknowns.

The system of equations is underdetermined (since each of them is consistent and the number of equations is larger than the number of variables) and so has infinitely many solutions.

Properties of splines S5

The simplest way of adding two more constraints is to require, in addition to the previous $3n-5$ constraints, that the spline $S(x)$ have an inflection point at each end of the defining interval $[x_1, x_n]$. The constraints added to Properties 1–3 are

Property 4a **Natural spline.** $S_1''(x_1) = 0$ and $S_{n-1}''(x_n) = 0$.

A cubic spline that satifies these two additional conditions is called a **natural** cubic spline. Note that (3.16) is a natural cubic spline, since it is easily verified from (3.18) that $S_1''(1) = 0$ and $S_3''(5) = 0$.

There are several other ways to add two more conditions. Usually, as in the case of the natural spline, they determine extra properties of the left and right ends of the spline, so they are called **end conditions**. We will take up this topic in the next section, but for now we concentrate on natural cubic splines.

Properties of splines S6

Now that we have the right number of equations, $3n - 3$ equations in $3n - 3$ unknowns, we can write a MATLAB function to solve them for the spline coefficients. First we write out the equations in the unknowns b_i, c_i, d_i. Part 2 of Property 1 then implies the $n - 1$ equations:

$$\begin{aligned} y_2 &= S_1(x_2) = y_1 + b_1(x_2 - x_1) + c_1(x_2 - x_1)^2 + d_1(x_2 - x_1)^3 \\ &\vdots \\ y_n &= S_{n-1}(x_n) = y_{n-1} + b_{n-1}(x_n - x_{n-1}) + c_{n-1}(x_n - x_{n-1})^2 \\ &\qquad + d_{n-1}(x_n - x_{n-1})^3. \end{aligned} \tag{3.19}$$

Property 2 generates the $n - 2$ equations,

$$\begin{aligned} 0 &= S_1'(x_2) - S_2'(x_2) = b_1 + 2c_1(x_2 - x_1) + 3d_1(x_2 - x_1)^2 - b_2 \\ &\vdots \\ 0 &= S_{n-2}'(x_{n-1}) - S_{n-1}'(x_{n-1}) = b_{n-2} + 2c_{n-2}(x_{n-1} - x_{n-2}) \\ &\qquad + 3d_{n-2}(x_{n-1} - x_{n-2})^2 - b_{n-1}, \end{aligned} \tag{3.20}$$

Properties of splines S7

and Property 3 implies the $n-2$ equations:

$$
\begin{aligned}
0 &= S_1''(x_2) - S_2''(x_2) = 2c_1 + 6d_1(x_2 - x_1) - 2c_2 \\
&\vdots \\
0 &= S_{n-2}''(x_{n-1}) - S_{n-1}''(x_{n-1}) = 2c_{n-2} + 6d_{n-2}(x_{n-1} - x_{n-2}) - 2c_{n-1}.
\end{aligned}
\tag{3.21}
$$

Instead of solving the equations directly, the system can be simplified by decoupling the equations.

The basic idea is to solve c_i first, and then write the formulas for the b_i and d_i in terms of the known c_i.

It is conceptually simpler if an extra unknown $c_n = S_{n-1}''(x_n)/2$ is introduced. In addition, we introduce the shorthand notation $\delta_i = x_{i+1} - x_i$ and $\Delta_i = y_{i+1} - y_i$. Then (3.21) can be solved for the coefficients

$$
d_i = \frac{c_{i+1} - c_i}{3\delta_i} \quad \text{for } i = 1, \ldots, n-1. \tag{3.22}
$$

Properties of splines S8

Solving (3.19) for b_i yields

$$\begin{aligned} b_i &= \frac{\Delta_i}{\delta_i} - c_i\delta_i - d_i\delta_i^2 \\ &= \frac{\Delta_i}{\delta_i} - c_i\delta_i - \frac{\delta_i}{3}(c_{i+1} - c_i) \\ &= \frac{\Delta_i}{\delta_i} - \frac{\delta_i}{3}(2c_i + c_{i+1}) \end{aligned} \tag{3.23}$$

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

Substituting (3.22) and (3.23) into (3.20) results in the following $n-2$ equations in $c_1, \ldots, c_n$:

$$\begin{aligned} \delta_1 c_1 + 2(\delta_1 + \delta_2)c_2 + \delta_2 c_3 &= 3\left(\frac{\Delta_2}{\delta_2} - \frac{\Delta_1}{\delta_1}\right) \\ &\vdots \\ \delta_{n-2}c_{n-2} + 2(\delta_{n-2} + \delta_{n-1})c_{n-1} + \delta_{n-1}c_n &= 3\left(\frac{\Delta_{n-1}}{\delta_{n-1}} - \frac{\Delta_{n-2}}{\delta_{n-2}}\right). \end{aligned}$$

Properties of splines S9

Two more equations are given by the natural spline conditions (Property 4a):

$$S_1''(x_1) = 0 \rightarrow 2c_1 = 0$$
$$S_{n-1}''(x_n) = 0 \rightarrow 2c_n = 0.$$

This gives a total of n equations in n unknowns c_i, which can be written in the matrix form

$$\begin{bmatrix} 1 & 0 & 0 & & & \\ \delta_1 & 2\delta_1 + 2\delta_2 & \delta_2 & \ddots & & \\ 0 & \delta_2 & 2\delta_2 + 2\delta_3 & \delta_3 & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & & \delta_{n-2} & 2\delta_{n-2} + 2\delta_{n-1} & \delta_{n-1} \\ & & & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ \\ \vdots \\ \\ c_n \end{bmatrix} = \begin{bmatrix} 0 \\ 3\left(\frac{\Delta_2}{\delta_2} - \frac{\Delta_1}{\delta_1}\right) \\ \vdots \\ 3\left(\frac{\Delta_{n-1}}{\delta_{n-1}} - \frac{\Delta_{n-2}}{\delta_{n-2}}\right) \\ 0 \end{bmatrix}. \quad (3.24)$$

After $c_1, \ldots, c_n$ are obtained from (3.24), $b_1, \ldots, b_{n-1}$ and $d_1, \ldots, d_{n-1}$ are found from (3.22) and (3.23).

Properties of splines S10

- Theorem

Let $n \geq 2$. For a set of data points $(x_1, y_1), \ldots, (x_n, y_n)$ with distinct x_i, there is a unique natural cubic spline fitting the points. ■

Natural cubic spline

Given $x = [x_1, \ldots, x_n]$ where $x_1 < \cdots < x_n$, $y = [y_1, \ldots, y_n]$

for $i = 1, \ldots, n-1$
$a_i = y_i$
$\delta_i = x_{i+1} - x_i$
$\Delta_i = y_{i+1} - y_i$
end
Solve (3.24) for $c_1, \ldots, c_n$
for $i = 1, \ldots, n-1$

$$d_i = \frac{c_{i+1} - c_i}{3\delta_i}$$

$$b_i = \frac{\Delta_i}{\delta_i} - \frac{\delta_i}{3}(2c_i + c_{i+1})$$

end
The natural cubic spline is
$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$ on $[x_i, x_{i+1}]$ for $i = 1, \ldots, n-1$.

Properties of splines S11

- Example

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

The x-coordinates are $x_1 = 0$, $x_2 = 1$, and $x_3 = 2$. The y-coordinates are $a_1 = y_1 = 3, a_2 = y_2 = -2$, and $a_3 = y_3 = 1$, and the differences are $\delta_1 = \delta_2 = 1$, $\Delta_1 = -5$, and $\Delta_2 = 3$. The tridiagonal matrix equation (3.24) is

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 4 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 24 \\ 0 \end{bmatrix}.$$

Properties of splines S12

- Example (cont…)

The solution is $[c_1, c_2, c_3] = [0, 6, 0]$. Now, (3.22) and (3.23) yield

$$d_1 = \frac{c_2 - c_1}{3\delta_1} = \frac{6}{3} = 2$$

$$d_2 = \frac{c_3 - c_2}{3\delta_2} = \frac{-6}{3} = -2$$

$$b_1 = \frac{\Delta_1}{\delta_1} - \frac{\delta_1}{3}(2c_1 + c_2) = -5 - \frac{1}{3}(6) = -7$$

$$b_2 = \frac{\Delta_2}{\delta_2} - \frac{\delta_2}{3}(2c_2 + c_3) = 3 - \frac{1}{3}(12) = -1.$$

Therefore, the cubic spline is

$$S_1(x) = 3 - 7x + 0x^2 + 2x^3 \text{ on } [0, 1]$$

$$S_2(x) = -2 - 1(x - 1) + 6(x - 1)^2 - 2(x - 1)^3 \text{ on } [1, 2].$$

splinecoeff.m splineplot.m testsplineplot.m

Endpoint conditions

Property 4b

Curvature-adjusted cubic spline. The first alternative to a natural cubic spline requires setting $S_1''(x_1)$ and $S_{n-1}''(x_n)$ to arbitrary values, chosen by the user, instead of zero. This choice corresponds to setting the desired curvatures at the left and right endpoints of the spline. In terms of (3.23), it translates to the two extra conditions

$$
\begin{aligned}
2c_1 &= v_1 \\
2c_n &= v_n,
\end{aligned}
$$

where v_1, v_n denote the desired values. The equations turn into the two tableau rows

$$
\left[\begin{array}{ccccccccc|c}
2 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & v_1 \\
0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 2 & v_n
\end{array}\right]
$$

to replace the top and bottom rows of (3.24), which were added for the natural spline.

Property 4c **Clamped cubic spline.**

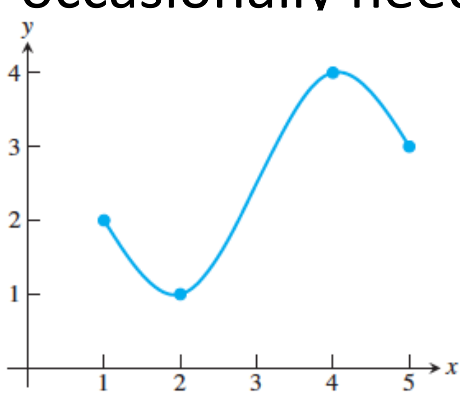
Property 4d **Parabolically terminated cubic spline.**

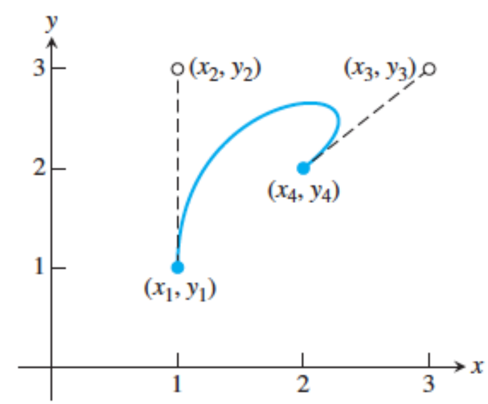
Property 4e **Not-a-knot cubic spline.**

Writing assignment:
Exercise 1, 5 @ page 176.
Coding assignment:
Computer problem 1 @ page 178.

Bezier Curves S1

- Recall that in cubic spline, each piece of splines are determined to as to ensure the smoothness of the first and second derivatives across the knot.
- Bezier curves are splines that allow the user to control the slopes at the knot with the loss of the smoothness of the first and second derivatives across the knot: suitable for cases where corners (discontinuous first derivatives) and abrupt changes in curvature (discontinuous second derivatives) are occasionally needed.





Bezier Curves S2

- History

Pierre Bezier 1959

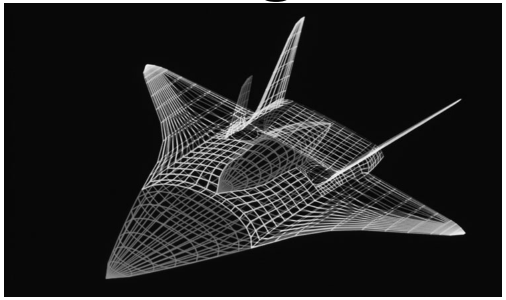
@ Renault automobile company

Paul de Casteljau 1958

@ Citroen automobile company

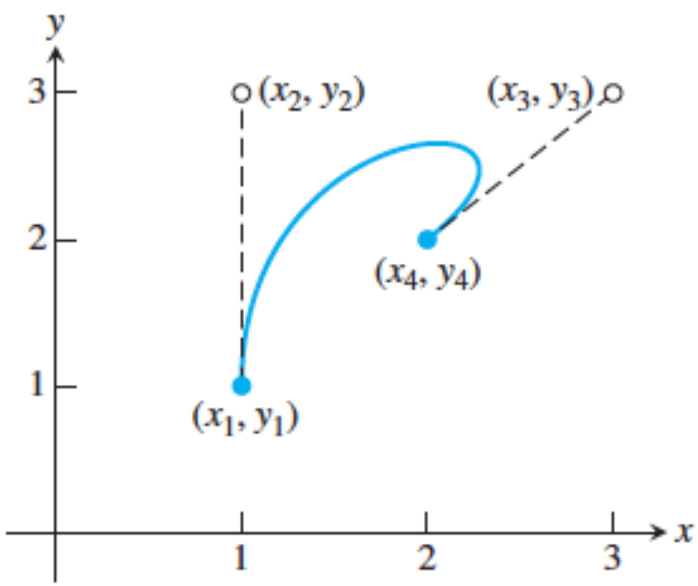
- Today

A cornerstone of computer-aided design and manufacturing.



Bezier Curves S3

- Basic idea: Each piece of a planar Bezier spline is determined by four points (x_1,y_1),(x_2,y_2), (x_3,y_3), (x_4,y_4) .
- End points: (x_1,y_1), (x_4,y_4); Control points: (x_2,y_2), (x_3,y_3)



The curve leaves

(x_1, y_1) along the tangent direction $(x_2 - x_1, y_2 - y_1)$ and ends at (x_4, y_4) along the tangent direction $(x_4 - x_3, y_4 - y_3)$. The equations that accomplish this are expressed as a parametric curve $(x(t), y(t))$ for $0 \le t \le 1$.

Bezier Curves S4

Bézier curve

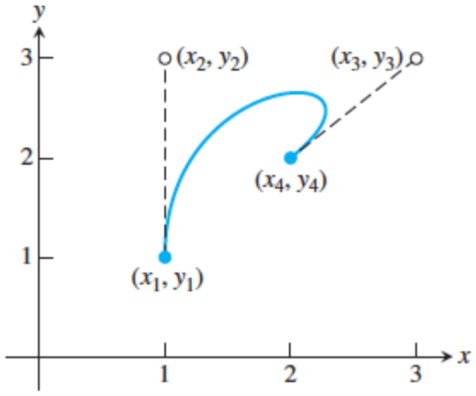
Given endpoints (x_1, y_1), (x_4, y_4)
control points (x_2, y_2), (x_3, y_3)

Set

$$b_x = 3(x_2 - x_1)$$
$$c_x = 3(x_3 - x_2) - b_x$$
$$d_x = x_4 - x_1 - b_x - c_x$$
$$b_y = 3(y_2 - y_1)$$
$$c_y = 3(y_3 - y_2) - b_y$$
$$d_y = y_4 - y_1 - b_y - c_y.$$

The Bézier curve is defined for $0 \le t \le 1$ by

$$x(t) = x_1 + b_x t + c_x t^2 + d_x t^3$$
$$y(t) = y_1 + b_y t + c_y t^2 + d_y t^3.$$



$$x(0) = x_1$$
$$x'(0) = 3(x_2 - x_1)$$
$$x(1) = x_4$$
$$x'(1) = 3(x_4 - x_3)$$

the analogous facts hold for $y(t)$.

Bezier Curves S5

- Why use these complicated formulas?

Linear Bezier curves through P_0 and P_1

$$\mathbf{B}(t) = \mathbf{P}_0 + t(\mathbf{P}_1 - \mathbf{P}_0) = (1-t)\mathbf{P}_0 + t\mathbf{P}_1 \ , t \in [0,1]$$

Quadratic Bezier curves through P_0, P_1 and P_2: A linear combination of the two linear Bezier curves.

$$\mathbf{B}(t) = (1-t)[(1-t)\mathbf{P}_0 + t\mathbf{P}_1] + t[(1-t)\mathbf{P}_1 + t\mathbf{P}_2] \ , t \in [0,1]$$

$$\Rightarrow \mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2 \ , t \in [0,1].$$

Cubic Bezier curves through P_0, P_1, P_2 and P_3: A linear combination of the two Quadratic Bezier curves.

$$\mathbf{B}(t) = (1-t)\mathbf{B}_{\mathbf{P}_0,\mathbf{P}_1,\mathbf{P}_2}(t) + t\mathbf{B}_{\mathbf{P}_1,\mathbf{P}_2,\mathbf{P}_3}(t) \ , t \in [0,1].$$

$$\Rightarrow \mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3 \ , t \in [0,1].$$

This formula explains why the Bezier formula shown in the previous slide works.

Bezier Curves S6

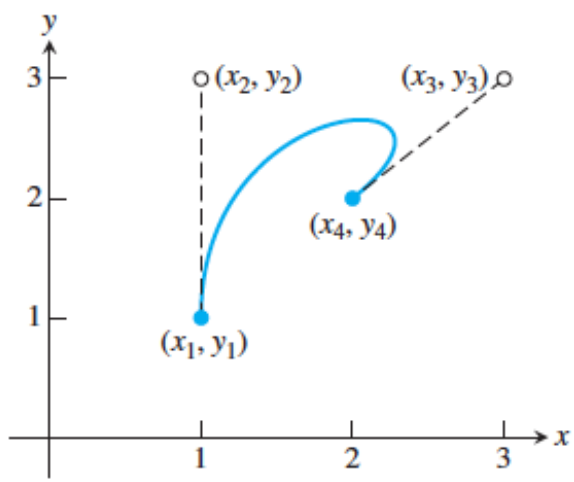
- Example

Find the Bézier curve $(x(t), y(t))$ through the points $(x, y) = (1, 1)$ and $(2, 2)$ with control points $(1, 3)$ and $(3, 3)$.

The four points are $(x_1, y_1) = (1, 1), (x_2, y_2) = (1, 3), (x_3, y_3) = (3, 3)$, and $(x_4, y_4) = (2, 2)$. The Bézier formulas yield $b_x = 0, c_x = 6, d_x = -5$ and $b_y = 6, c_y = -6, d_y = 1$. The Bézier spline

$$x(t) = 1 + 6t^2 - 5t^3$$
$$y(t) = 1 + 6t - 6t^2 + t^3$$

is shown in Figure 3.14 along with the control points.



Bezier Curves S7

- Example

Prove that the Bézier spline with $(x_1, y_1) = (x_2, y_2)$ and $(x_3, y_3) = (x_4, y_4)$ is a line segment.

The Bézier formulas show that the equations are

$$x(t) = x_1 + 3(x_4 - x_1)t^2 - 2(x_4 - x_1)t^3 = x_1 + (x_4 - x_1)t^2(3 - 2t)$$
$$y(t) = y_1 + 3(y_4 - y_1)t^2 - 2(y_4 - y_1)t^3 = y_1 + (y_4 - y_1)t^2(3 - 2t)$$

for $0 \leq t \leq 1$. Every point in the spline has the form

$$\begin{aligned}(x(t), y(t)) &= (x_1 + r(x_4 - x_1), y_1 + r(y_4 - y_1)) \\ &= ((1 - r)x_1 + rx_4, (1 - r)y_1 + ry_4),\end{aligned}$$

where $r = t^2(3 - 2t)$. Since $0 \leq r \leq 1$, each point lies on the line segment connecting (x_1, y_1) and (x_4, y_4). ◀

Writing assignment: Exercise 1 @ page 182
Coding assignment: Computer problem 1 @ page 183.

Bezier Curves: Project Two

- Input: You scanned signature (in image format, e.g., .jpg, .bmp).
- Output: The signature generated by Bezier curves, in PDF format.
- You should submit: scanned/Bezier signature, all the codes, and a report on how you do it.
- All the basic materials can be found in Reality Check 3 and Program 3.7 in the text book.
- Demo: My scanned signature of “王”.
- Modifiedbezierdraw.m