MA 214: Introduction to numerical analysis
Lecture 17

Shripad M. Garge.
IIT Bombay

(shripad@math.iitb.ac.in)

2021-2022

Piecewise polynomial interpolation

We have been studying interpolations by a single polynomial.

However, high-degree polynomials can oscillate erratically, a minor fluctuation over a small portion of the interval can induce large fluctuations over the entire range.

An alternative approach is to divide the approximation interval into a collection of subintervals and construct a (generally) different approximating polynomial on each sub-interval.

This is called piecewise-polynomial interpolation or a spline.

Linear spline

The simplest spline is the linear spline, which consists of joining a set of data points

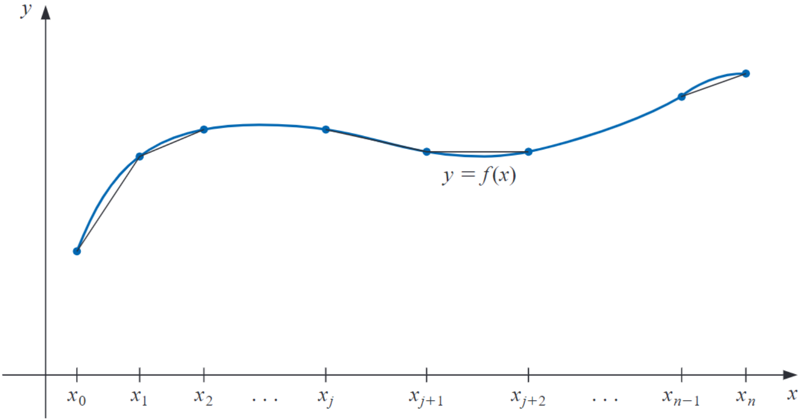
$$\Big\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\Big\}$$

by a series of straight lines.

A disadvantage of linear spline is that there is likely no differentiability at the endpoints of the subintervals, which means that the interpolating function is not "smooth".

Often it is clear from physical conditions that smoothness is required, so the spline must be continuously differentiable.

Linear spline



Quadratic spline?

The simplest differentiable spline on an entire interval $[x_0, x_n]$ is the function obtained by fitting a quadratic polynomial between each successive pair of nodes.

This is done by constructing a quadratic on $[x_0, x_1]$ agreeing with the function at x_0 and x_1, another quadratic on $[x_1, x_2]$ agreeing with the function at x_1 and x_2, and so on.

A general quadratic polynomial has three arbitrary constants;

- the constant term,
- the coefficient of x, and
- the coefficient of x^2.

Further, only two conditions are required to fit the data at the endpoints of each subinterval.

Quadratic spline?

So we have the flexibility to permit the quadratics to be chosen so that the interpolant has a continuous derivative on $[x_0, x_n]$.

The difficulty arises because we generally need to specify conditions about the derivative of the interpolant at the endpoints x_0 and x_n.

There is not a sufficient number of constants to ensure that the conditions will be satisfied.

Let us try to understand this in the simplest case.

Quadratic spline?

Suppose we have three nodes, $a = x_0 < x_1 < x_2 = b$. A quadratic spline S consists of quadratic polynomials:

- $S_0(x) = a_0 + b_0(x - x_0) + c_0(x - x_0)^2$ on $[x_0, x_1]$

and

- $S_1(x) = a_1 + b_1(x - x_1) + c_1(x - x_1)^2$ on $[x_1, x_2]$.

We then have $a_0 = f(x_0), a_1 = f(x_1)$ and $b_1 = f'(x_1)$.

Further, $S_0(x_1) = f(x_1), S_0'(x_1) = f'(x_1)$ and $S_1(x_2) = f(x_2)$ determine the remaining quantities. Thus, the quadratic spline S is uniquely determined.

Then, what about $S'(x_0) = f'(x_0)$ and $S'(x_2) = f'(x_2)$?

Cubic spline

The most common piecewise-polynomial interpolation uses cubic polynomials between each successive pair of nodes and is called a cubic spline.

A general cubic polynomial involves four constants, so there is sufficient flexibility in the cubic spline procedure to ensure that the interpolant is not only continuously differentiable on the interval, but also has a continuous second derivative.

The construction of the cubic spline does not, however, assume that the derivatives of the interpolant agree with those of the function it is approximating, even at the nodes.

Cubic spline

Given a function f defined on $[a, b]$ and a set of nodes $a = x_0 < x_1 < \cdots < x_n = b$, a cubic spline interpolant S for f is a function that satisfies the following conditions:

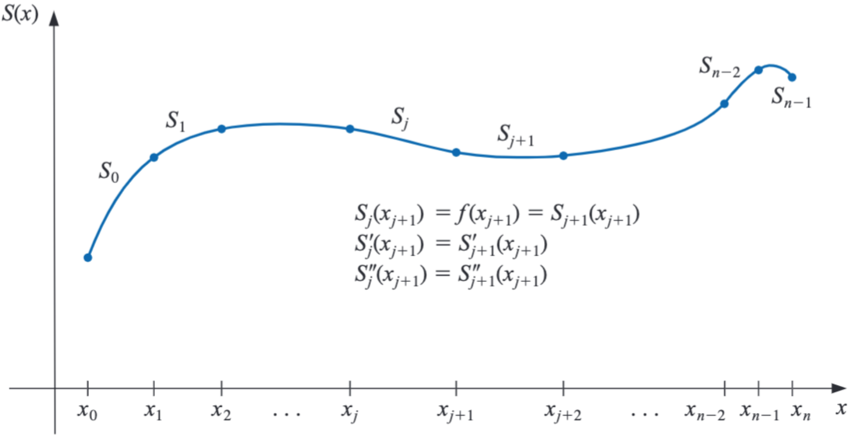
1. $S(x)$ is a cubic polynomial $S_j(x)$ on each $[x_j, x_{j+1}]$;
2. $S_j(x_j) = f(x_j)$ and $S_j(x_{j+1}) = f(x_{j+1})$;
3. $S_j'(x_{j+1}) = S_{j+1}'(x_{j+1})$ and $S_j''(x_{j+1}) = S_{j+1}''(x_{j+1})$.

There are two types of boundary conditions that we may impose:

(i) $S''(x_0) = S''(x_n) = 0$, natural (or free) boundary;

(ii) $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$, clamped boundary.

Cubic spline



MA 214: Introduction to numerical analysis
Lecture 18

Shripad M. Garge.
IIT Bombay

(shripad@math.iitb.ac.in)

2021-2022

Cubic spline

Let $f : [a, b] \to \mathbb{R}$ be a function and $a = x_0 < x_1 < \cdots < x_n = b$ be distinct nodes. A cubic spline interpolant S for f is a function that satisfies:

1. $S(x) = S_j(x)$ is a cubic polynomial on each $[x_j, x_{j+1}]$;
2. $S_j(x_j) = f(x_j)$ and $S_j(x_{j+1}) = f(x_{j+1}) = S_{j+1}(x_{j+1})$;
3. $S_j'(x_{j+1}) = S_{j+1}'(x_{j+1})$ and $S_j''(x_{j+1}) = S_{j+1}''(x_{j+1})$.

Boundary conditions:

(i) Natural or free boundary: $S''(x_0) = S''(x_n) = 0$;

(ii) Clamped boundary: $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$.

Cubic spline

It is more common to have natural cubic splines than the clamped ones, simply because the information about f' at various points may not always be available.

But whenever this information is available, we get a better approximation for the function f on the interval $[a, b]$ using the clamped boundary conditions.

Before going on for the general construction of cubic splines, we will see an example which will illustrate the general case.

An example

Let us construct a natural cubic spline passing through the points $(1, 2)$, $(2, 3)$ and $(3, 5)$.

Here we have two cubics:

$S_0(x) = a_0 + b_0(x-1) + c_0(x-1)^2 + d_0(x-1)^3$ on $[1, 2]$ and

$S_1(x) = a_1 + b_1(x-2) + c_1(x-2)^2 + d_1(x-2)^3$ on $[2, 3]$.

We have $a_0 = S_0(1) = f(1) = 2$, $a_1 = S_1(2) = f(2) = 3$, and

$S_0(2) = a_0 + b_0 + c_0 + d_0 = 3$, $\quad S_1(3) = a_1 + b_1 + c_1 + d_1 = 5$.

By $S_0'(2) = S_1'(2)$ and $S_0''(2) = S_1''(2)$ we get

$$b_0 + 2c_0 + 3d_0 = b_1 \quad \text{and} \quad 2c_0 + 6d_0 = 2c_1.$$

The example, continued

Finally, we impose the natural boundary conditions: $S_0''(1) = 0$ and $S_1''(3) = 0$ to get

$$2c_0 = 0 \quad \text{and} \quad 2c_1 + 6d_1 = 0.$$

Solving all these equations gives us the spline

$$S(x) = \begin{cases} 2 + \frac{3}{4}(x-1) + \frac{1}{4}(x-1)^3 & x \in [1,2], \\ 3 + \frac{3}{2}(x-2) + \frac{3}{4}(x-2)^2 - \frac{1}{4}(x-2)^3 & x \in [2,3]. \end{cases}$$

The general construction of a cubic spline

As the example demonstrates, a cubic spline defined on an interval that is divided into n subintervals will require determining $4n$ constants.

To construct the cubic spline interpolant for a given function f, the conditions in the definition are applied to the cubic polynomials

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

for $j = 0, 1, \dots, n-1$.

Since $S_j(x_j) = a_j = f(x_j)$ and $S_j(x_{j+1}) = S_{j+1}(x_{j+1})$, we get

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

With $h_j := (x_{j+1} - x_j)$ we get

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3. \tag{1}$$

Similarly, we have $b_j = S_j'(x_j)$ and $S_j'(x_{j+1}) = S_{j+1}'(x_{j+1})$ which gives

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2 \qquad (2)$$

and $2c_j = S_j''(x_j)$, $S_j''(x_{j+1}) = S_{j+1}''(x_{j+1})$ to give

$$c_{j+1} = c_j + 3d_j h_j. \qquad (3)$$

With $a_n = f(x_n) = S(x_n)$, $b_n = S'(x_n)$ and $2c_n = S''(x_n)$, these equations are valid for $j = 0, 1, \ldots, n-1$. We solve for d_j in equation (3) in terms of c_j and c_{j+1}, and put it in (1) and (2):

$$\begin{aligned} a_{j+1} &= a_j + b_j h_j + \frac{h_j^2}{3}(2c_j + c_{j+1}), & (4)\\ b_{j+1} &= b_j + h_j(c_j + c_{j+1}). & (5) \end{aligned}$$

We then solve (4) for b_j in terms of a_j, c_j and h_j to get

$$b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}). \tag{6}$$

Finally, (5) and (6) together give

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1})$$

$$\ldots \quad (7)$$

for $j = 1, 2, \ldots, n-1$.

In this system, h_j and a_j are known while c_j are unknown. Once the c_j are determined, we trace the above analysis backwards and find the values of b_j and d_j to determine the cubic polynomials S_j.

Once we impose one of the boundary conditions, natural or clamped, the above system can be solved for c_j.

The proof of the above statement, which we will not give here in complete detail, involves linear algebra.

We will illustrate it by means of an example.

Consider the data (x, e^x), for $x = 0, 1, 2, 3$. We then have $n = 3$, $h_0 = h_1 = h_2 = 1$, $a_0 = 1$, $a_1 = e$, $a_2 = e^2$ and $a_3 = e^3$.

We then get

$$c_0 = 0, \qquad c_0 + 4c_1 + c_2 = 3(e^2 - 2e + 1),$$

$$c_1 + 4c_2 + c_3 = 3(e^3 - 2e^2 + e), \qquad c_3 = 0.$$

Another example

This translates to the matrix equation

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 3(e^2 - 2e + 1) \\ 3(e^3 - 2e^2 + e) \\ 0 \end{pmatrix}.$$

This system is solvable and gives $c_0 = 0$, $c_3 = 0$,

$$c_1 = \frac{1}{5}(-e^3 + 6e^2 - 9e + 4) \approx 0.7568$$

$$c_2 = \frac{1}{5}(4e^3 - 9e^2 + 6e - 1) \approx 5.83007.$$

We can put these values in above equations to solve for b_j and d_j.

The example continued

The natural cubic spline then is given by

$$S_0(x) = 1 + 1.46600x + 0.25228x^3,$$

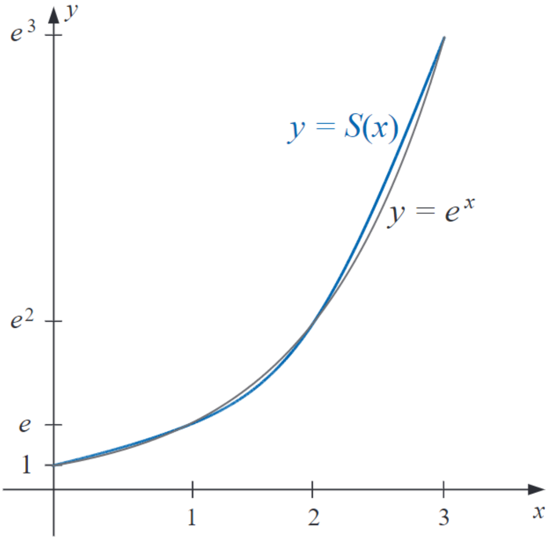
$$S_1(x) = 2.71828+2.22285(x-1)+0.75685(x-1)^2+1.69107(x-1)^3,$$

$$S_2(x) = 7.38906+8.80977(x-2)+5.83007(x-2)^2-1.94336(x-2)^3.$$

One application of this computation is that

$$\int_0^3 e^x dx \approx \int_0^1 S_0(x)dx + \int_1^2 S_1(x) + \int_2^3 S_2(x).$$

The example continued



MA 214: Introduction to numerical analysis
Lecture 19

Shripad M. Garge.
IIT Bombay

(shripad@math.iitb.ac.in)

2021-2022

An example of clamped spline

In the previous example, we found a natural spline S that passes through the points $(1, 2)$, $(2, 3)$ and $(3, 5)$.

Here we construct a clamped splines through these points that has $S'(1) = 2$ and $S'(3) = 1$. Let

$$S_0(x) = a_0 + b_0(x-1) + c_0(x-1)^2 + d_0(x-1)^3$$

be the cubic on $[1, 2]$ and

$$S_1(x) = a_1 + b_1(x-2) + c_1(x-2)^2 + d_1(x-2)^3$$

be the cubic on $[2, 3]$.

Then most of the conditions to determine the 8 constants are the same as those in the previous example.

The example, continued

We have $f(1) = a_0 = 2$, $f(2) = a_0 + b_0 + c_0 + d_0 = a_1 = 3$, and $f(3) = a_1 + b_1 + c_1 + d_1 = 5$.

Further, $S_0'(2) = S_1'(2)$ and $S_0''(2) = S_1''(2)$ give

$$b_0 + 2c_0 + 3d_0 = b_1 \quad \text{and} \quad 2c_0 + 6d_0 = 2c_1.$$

The boundary conditions are new:

$$S_0'(1) = b_0 = 2 \quad \text{and} \quad S_1'(3) = b_1 + 2c_1 + 3d_1 = 1.$$

We get the clamped spline as

$$S(x) = \begin{cases} 2 + 2(x-1) - \frac{5}{2}(x-1)^2 + \frac{3}{2}(x-1)^3, & x \in [1,2], \\ 3 + \frac{3}{2}(x-2) + 2(x-2)^2 - \frac{3}{2}(x-2)^3, & x \in [2,3]. \end{cases}$$

Existence of splines

Just as the natural spline exists, we have the clamped spline as well. That is, the set of equations obtained by the clamped boundary conditions leads to a unique solution.

This proof also involves linear algebra and we will skip it. The statements read as follows:

If f is defined at $a = x_0 < x_1 < \cdots < x_n = b$ then f has a unique natural spline interpolant S on the nodes $x_0, x_1, \ldots, x_n$.

If f is defined at $a = x_0 < x_1 < \cdots < x_n = b$ and is differentiable at a and b, then f has a unique clamped spline interpolant S on the nodes $x_0, x_1, \ldots, x_n$.

Another example

We will repeat our second example of $(x, f(x))$:

$$(0, e^0), \ (1, e), \ (2, e^2) \quad \text{and} \quad (3, e^3);$$

with the clamped boundary conditions $f'(0) = 1$ and $f'(3) = e^3$.

We have the cubic polynomials

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

on $[x_j, x_{j+1}]$ for $j = 0, 1, 2$.

As in the earlier instance of this example, we have $n = 3$, $h_j = 1$, $a_0 = 1$, $a_1 = e$, $a_2 = e^2$ and $a_3 = e^3$.

This together with the information that $f'(0) = 1$ and $f'(3) = e^3$ gives the matrix equation:

The example, continued

$$\begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 3(e-2) \\ 3(e^2-2e+1) \\ 3(e^3-2e^2+e) \\ 3e^2 \end{pmatrix}$$

Solving this system gives

$$c_0 \approx 0.44468, \quad c_1 \approx 1.26548, \quad c_2 \approx 3.35087 \quad \text{and} \quad c_3 \approx 9.40815.$$

We use these values to compute the remaining constants which further give the required cubic polynomials:

$$1 + x + 0.44468x^2 + 0.27360x^3,$$

$$2.71828 + 2.71016(x-1) + 1.26548(x-1)^2 + 0.69513(x-1)^3$$

$$7.38906 + 7.32652(x-2) + 3.35087(x-2)^2 + 2.01909(x-2)^3.$$

Natural vis-à-vis clamped cubic spline

We indicated an application of the cubic spline in the computation of the integral

$$\int_0^3 e^x\, dx.$$

The actual value of this integral is 19.08554.

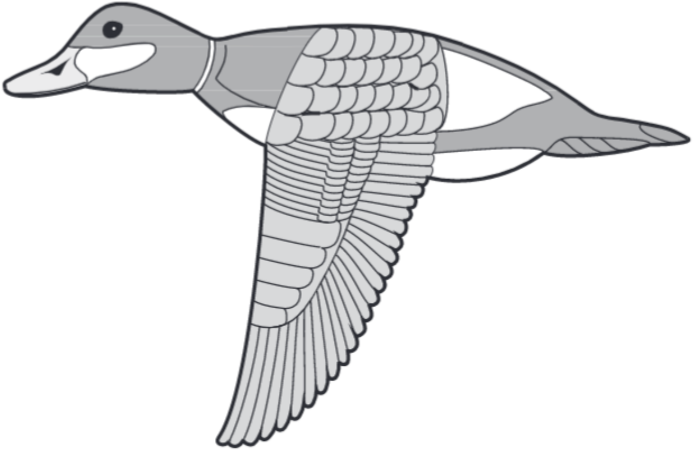
The natural cubic spline gives the value 19.55229 while the clamped cubic spline gives 19.05965.

We immediately see that the clamped spline is superior.

This is no surprise as the natural boundary conditions are $S''(0) = S_2''(3) = 0$ while the actual values are $f''(0) = 1$ and $f''(3) = e^3$.

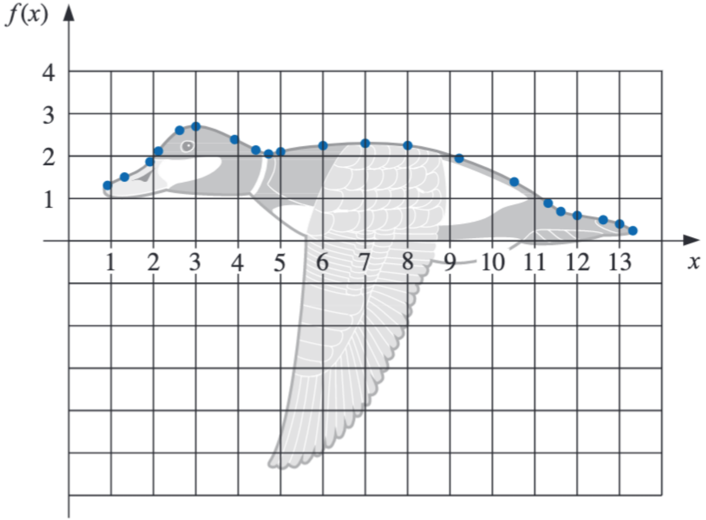
One more example

We can use cubic splines in the scenario where we have no description of the underlying function. Let us say that we want to approximate the top profile of our flying duck.



The example, continued

We choose 21 data points to approximate the top profile.



The example, continued

The nodes and the corresponding function values are

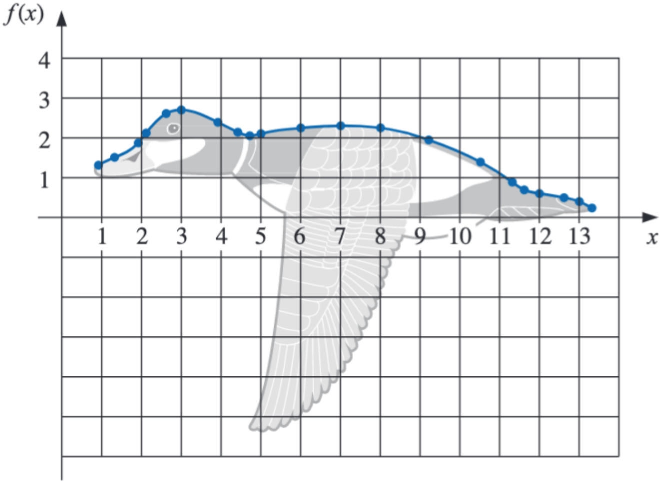
x	0.9	1.3	1.9	2.1	2.6	3.0	3.9
$f(x)$	1.3	1.5	1.85	2.1	2.6	2.7	2.4
x	4.4	4.7	5.0	6.0	7.0	8.0	9.2
$f(x)$	2.15	2.05	2.1	2.25	2.3	2.25	1.95
x	10.5	11.3	11.6	12.0	12.6	13.0	13.3
$f(x)$	1.4	0.9	0.7	0.6	0.5	0.4	0.25

We compute the natural cubic spline for this data.

The example, continued

j	x_j	a_j	b_j	c_j	d_j
0	0.9	1.3	5.40	0.00	−0.25
1	1.3	1.5	0.42	−0.30	0.95
2	1.9	1.85	1.09	1.41	−2.96
3	2.1	2.1	1.29	−0.37	−0.45
4	2.6	2.6	0.59	−1.04	0.45
5	3.0	2.7	−0.02	−0.50	0.17
6	3.9	2.4	−0.50	−0.03	0.08
7	4.4	2.15	−0.48	0.08	1.31
8	4.7	2.05	−0.07	1.27	−1.58
9	5.0	2.1	0.26	−0.16	0.04
10	6.0	2.25	0.08	−0.03	0.00
11	7.0	2.3	0.01	−0.04	−0.02
12	8.0	2.25	−0.14	−0.11	0.02
13	9.2	1.95	−0.34	−0.05	−0.01
14	10.5	1.4	−0.53	−0.10	−0.02
15	11.3	0.9	−0.73	−0.15	1.21
16	11.6	0.7	−0.49	0.94	−0.84
17	12.0	0.6	−0.14	−0.06	0.04
18	12.6	0.5	−0.18	0.00	−0.45
19	13.0	0.4	−0.39	−0.54	0.60
20	13.3	0.25			

This spline curve is nearly identical to the profile, as shown in the following figure.



The example, continued

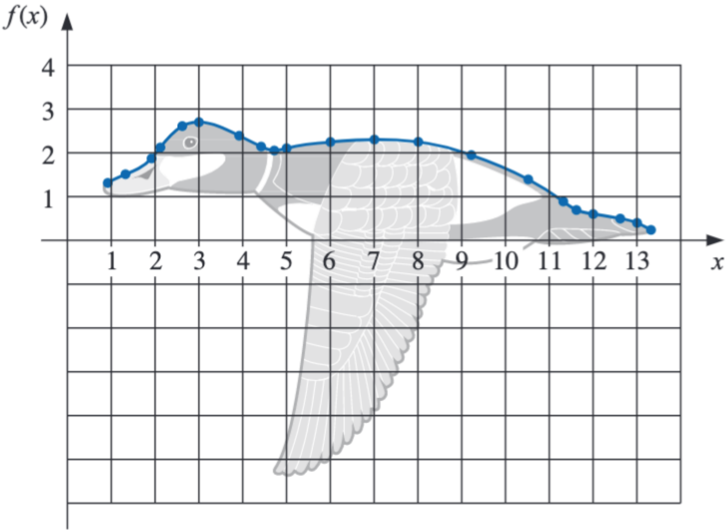
To use a clamped spline to approximate this curve we would need derivative approximations for the endpoints.

Even if these approximations were available, we could expect little improvement because of the close agreement of the natural cubic spline to the curve of the top profile.

Constructing a cubic spline to approximate the lower profile of the duck would be more difficult since the curve for this portion cannot be expressed as a function of x, and at certain points the curve does not appear to be smooth.

Such a situation requires another approach but we will not study it in our course.

The example, continued



Error

Let f be 4 times continuously differentiable on $[a, b]$ with

$$\max_{a \leqslant x \leqslant b} |f^{(4)}(x)| = M.$$

If S is the clamped cubic spline interpolant to f with respect to the nodes $a = x_0 < x_1 < \cdots < x_n = b$, then for all $x \in [a, b]$,

$$|f(x) - S(x)| \leqslant \frac{5M}{384} \max_j (x_{j+1} - x_j)^4.$$

A fourth-order error-bound result also holds in the case of natural boundary conditions, but it is more difficult to express.

This is because the natural cubic spline is more error-prone than the clamped ones.

The third theme is complete

This completes the third theme of our course:

Interpolation.

From our next lecture, we will begin the fourth theme:

Numerical Differentiation and Integration.

MA 214: Introduction to numerical analysis
Lecture 20

Shripad M. Garge.
IIT Bombay

(shripad@math.iitb.ac.in)

2021-2022

We begin our fourth theme "Numerical Differentiation and Integration".

Let us start with a potential application.

A sheet of corrugated roofing is constructed by pressing a flat sheet of aluminium into one whose cross section has the form of a sine wave.

A potential application

A corrugated sheet 4 ft long is needed, the height of each wave is 1 in. from the centerline, and each wave has a period of approximately 2π in.

The problem of finding the length of the initial flat sheet is one of determining the length of the curve given by $f(x) = \sin x$ from $x = 0$ in. to $x = 48$ in.

From calculus, we know this length is

$$\int_0^{48} \sqrt{1 + (f'(x))^2}dx = \int_0^{48} \sqrt{1 + \cos^2 x}dx.$$

Computing this involves an elliptic integral of the second kind, which cannot be evaluated explicitly. So we need to develop numerical methods to compute it.

Previous themes help us

We used polynomials to approximate an arbitrary set of data because polynomials approximate continuous function uniformly on a closed interval.

Also, the derivatives and integrals of polynomials are easily obtained and evaluated.

It should not be surprising, then, that many procedures for approximating derivatives and integrals use the polynomials that approximate the function.

Approximating $f'(x)$

The derivative of f at x_0 is

$$f'(x_0) = \lim_{h\to 0} \frac{f(x_0+h) - f(x_0)}{h}$$

so at first $f'(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h}$ for small enough h.

But, we would also like to know the error. Let us begin from the beginning then.

Let us assume that our f is twice continuously differentiable and let P be the linear polynomial interpolating f on x_0 and x_0+h for a small enough h:

$$f(x) = \frac{f(x_0)(x-x_0-h)}{-h} + \frac{f(x_0+h)(x-x_0)}{h} + \frac{(x-x_0)(x-x_0-h)}{2} f''(\xi(x)).$$

Approximating $f'(x)$

Differentiating the above equation gives

$$f'(x) = \frac{f(x_0+h) - f(x_0)}{h} + \frac{2(x-x_0)-h}{2} f''(\xi(x))$$

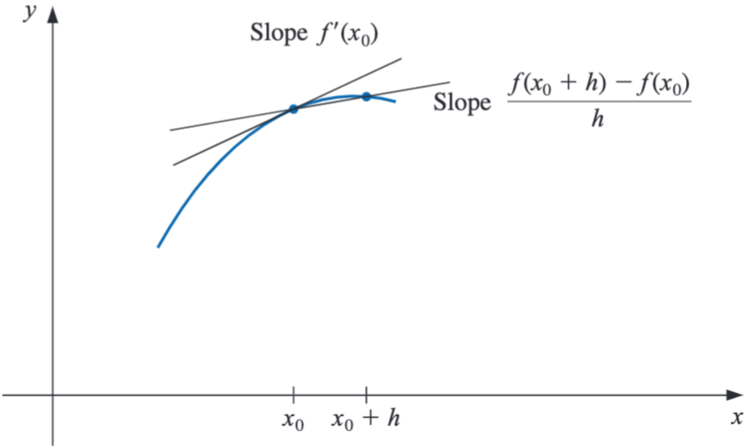
$$+ \frac{(x-x_0)(x-x_0-h)}{2} \frac{d}{dx}\left(f''(\xi(x))\right).$$

Then $x = x_0$ gives

$$f'(x_0) = \frac{f(x_0+h) - f(x_0)}{h} - \frac{h}{2} f''(\xi(x))$$

For small enough h, $[f(x_0+h) - f(x_0)]/h$ is an approximation to $f'(x_0)$ with an error $\leqslant M|h|/2$, where M is an upper bound on $|f''(x)|$ on $[x_0, x_0+h]$.

Approximating $f'(x)$



If $h > 0$, the formula is called the forward-difference formula, else the backward-difference formula.

An example

Let $f(x) = \ln(x)$ and $x_0 = 1.8$. The forward-difference formula is

$$f'(1.8) \approx \frac{f(1.8+h) - f(1.8)}{h}.$$

With $h = 0.1$ the approximation is

$$\frac{\ln(1.9) - \ln(1.8)}{0.1} = 0.5406722127.$$

A bound for the corresponding error is

$$\frac{|hf''(\xi)|}{2} = \frac{|h|}{2\xi^2} < \frac{0.1}{2(1.8)^2} = 0.01543209876.$$

We next give a table for a few such values of h.

The example, continued

h	$f(1.8+h)$	$\dfrac{f(1.8+h)-f(1.8)}{h}$	$\dfrac{\lvert h\rvert}{2(1.8)^2}$
0.1	0.64185388617	0.5406722127	0.01543209876
0.08	0.63127177684	0.54356389924	0.01234567901
0.05	0.61518563909	0.54797948376	0.00771604938
0.02	0.59883650108	0.55249180932	0.00308641975
0.01	0.59332684527	0.55401803756	0.00154320987

We know the value of $f'(x) = 1/x$ at 1.8 to be $0.5555\ldots$ which is reasonably approximated at $h = 0.01$.

This encourages us to use interpolating polynomials at a higher number of nodes.

Use interpolating polynomials

Let $x_0, \ldots, x_n$ be distinct nodes in an interval I and let f be $(n+1)$ times continuously differentiable on I.

Then for $x \in I$

$$f(x) = \sum_{k=0}^{n+1} f(x_k) L_k(x) + \frac{(x - x_0)\cdots(x - x_n)}{(n+1)!} f^{(n+1)}(\xi(x)).$$

Differentiating this formula and putting $x = x_j$ we get

$$f'(x_j) = \sum_{k=0}^{n+1} f(x_k) L_k'(x_j) + \frac{f^{(n+1)}(\xi(x_j))}{(n+1)!} \prod_{k \neq j} (x_j - x_k).$$

This is called the $(n+1)$-point formula to approximate $f'(x_j)$.

Three point formula

When $n = 2$, we get the 3-point formula:

$$\begin{aligned} f'(x_j) &= f(x_0)\left[\frac{2x_j - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)}\right] \\ &\quad + f(x_1)\left[\frac{2x_j - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)}\right] \\ &\quad + f(x_2)\left[\frac{2x_j - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)}\right] \\ &\quad + \frac{f^{(3)}(\xi_j)}{6}\prod_{k \neq j}(x_j - x_k). \end{aligned}$$

If we let $x_1 = x_0 + h$ and $x_2 = x_0 + 2h$ then the above formula becomes simpler.

Three point formula

In particular,

$$
\begin{aligned}
f'(x_0) &= \frac{1}{h}\left[-\frac{3}{2}f(x_0) + 2f(x_1) - \frac{1}{2}f(x_2)\right] + \frac{h^2}{3}f^{(3)}(\xi_0) \\
f'(x_1) &= \frac{1}{h}\left[-\frac{1}{2}f(x_0) + \frac{1}{2}f(x_2)\right] - \frac{h^2}{6}f^{(3)}(\xi_1) \\
f'(x_2) &= \frac{1}{h}\left[\frac{1}{2}f(x_0) - 2f(x_1) + \frac{3}{2}f(x_2)\right] + \frac{h^2}{3}f^{(3)}(\xi_2)
\end{aligned}
$$

Note the symmetry in the equations for $f'(x_0)$ and $f'(x_2)$.

If we write these formulae assuming that the point x_j is the point x_0, and adjusting the other points with the help of h then we get essentially two formulae.

Three point formula

Three-point endpoint formula:

$$f'(x_0) = \frac{1}{2h}\Big(-3f(x_0) + 4f(x_0+h) - f(x_0+2h)\Big) + \frac{h^2}{3} f^{(3)}(\xi_0)$$

where ξ_0 lies between x_0 and $x_0 + 2h$.

Three-point midpoint formula:

$$f'(x_0) = \frac{1}{2h}\Big(f(x_0+h) - f(x_0-h)\Big) - \frac{h^2}{6} f^{(3)}(\xi_0)$$

where ξ_0 lies between $x_0 - h$ and $x_0 + h$.

Both the errors are of the order of h^2 but the second error is about half of the first error.

Three point formula